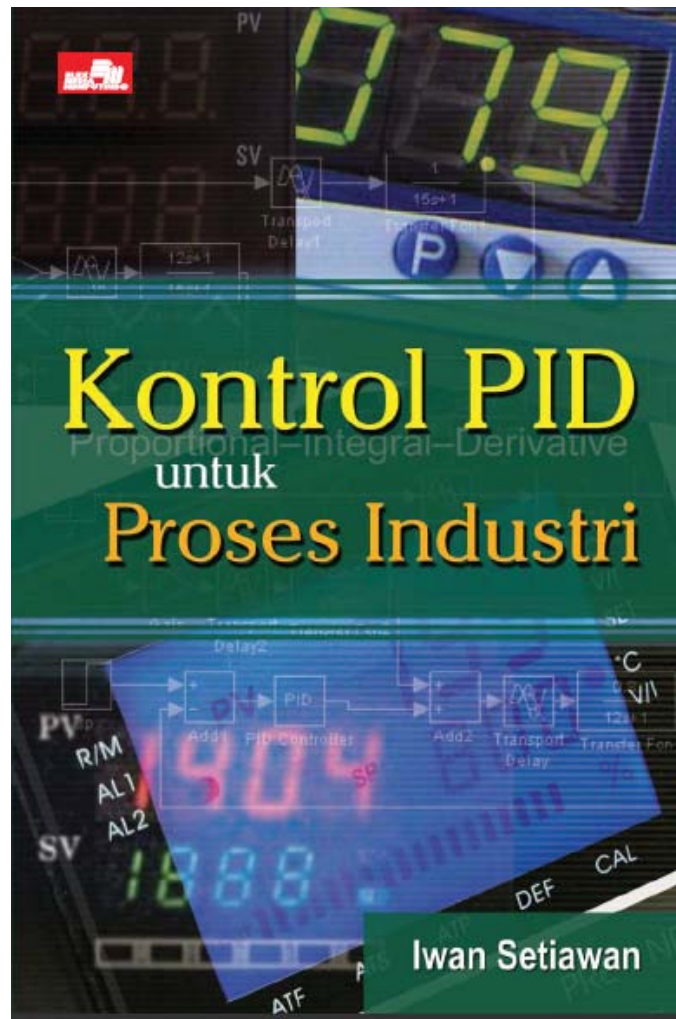


# KONTROL PID UNTUK PROSES INDUSTRI

Beragam Struktur dan Metode Tuning PID praktis

(Iwan Setiawan)

Cover Buku Asli:



Diterbitkan pertama kali oleh Penerbit PT Elex Media Komputindo,  
Tahun 2008

Dengan ISBN: 978-979-27-4100-1

## *Untuk*

*Istriku Ria dan Anak-anakku Muhammad Naufal R. Setiawan  
dan Muhammad Raditya D. Setiawan  
Serta Bapak (alm) dan Ibu*



**Iwan Setiawan**, Lahir di Serang, Banten 26 September 1973. Bekerja sebagai dosen tetap di jurusan Teknik Elektro Undip, Semarang. Menyelesaikan Studi Sarjana dan Master berturut-turut pada tahun 1998 dan 2003 di Teknik Elektro UGM, Yogyakarta dengan bidang keminatan Sistem Kontrol dan Instrumentasi. Sejak tahun 2012 menempuh pendidikan tingkat Doktor di Teknik Elektro ITS, Surabaya. Bidang yang penulis dalam terutama desain sistem kontrol inverter/konverter untuk aplikasi pembangkitan daya listrik berbasis energi terbarukan.

Penulis dapat dihubungi via E-mail:  
[setiaone.iwan@gmail.com](mailto:setiaone.iwan@gmail.com)

## Kata Pengantar Cetakan Bebas

Hampir beberapa tahun terakhir ini, penulis sering mendapatkan e-mail dari Mahasiswa dan Karyawan yang menanyakan keberadaan **buku Kontrol PID untuk Proses Industri** yang penulis susun dengan maksud untuk mendapatkan/membeli buku tersebut dari penulis secara pribadi (hal ini dikarenakan buku ini nampaknya sudah tidak tersedia lagi di toko-toko buku dunia nyata maupun dunia maya).

Dengan niatan untuk bisa saling berbagi, maka penulis berinisiatif menghubungi pihak Penerbit **PT. Elex Media Komputindo** untuk meminta izin menerbitkan draft buku Kontrol PID untuk Proses Industri secara bebas-gratis (hal ini dilakukan dengan mempertimbangkan prinsip kehati-hatian--agar tidak melanggar hak Cipta),). Alhamdulillah penulis akhirnya mendapatkan izin yang dimaksud.

Penulis berharap semoga usaha ini mendapatkan ridho Allah SWT dan buku ini bisa memberi banyak manfaat bagi para pembaca.

Terakhir penulis mengucapkan banyak terimakasih pada PT. Elex Media Komputindo yang telah mengizinkan penerbitan buku ini secara bebas.

Surabaya, Mei 2015

Penulis

## **Surat Izin Cetakan Bebas dari pihak Penerbit PT. Elex Media Komputindo**

Yth. Bp. Iwan Setiawan,

Menanggapi permohonan bapak di bawah, kami setuju dan mengizinkan bapak menerbitkan draft buku tsb secara gratis di internet.

Hal ini karena buku tsb memang sudah tidak ada lagi stoknya dan kami tidak berencana mencetak ulang buku tsb.

Demikian pak jawaban kami. Terima kasih atas perhatiannya.

Salam,

V. Sugeng Hardojo  
Non Fiction Editorial Manager  
PT Elex Media Komputindo

## **Surat dari Penulis:**

Kepada Yth. Dewan redaksi PT Elex Media Komputindo

Sebelumnya perkenalkan nama saya Iwan setiawan, dosen Teknik Elektro Undip sebagai penulis buku Kontrol PID untuk Proses industri (yang diterbitkan PT. Elex pada tahun 2008, dg. ISBN 978-979-27-4100-1).

Terkait dengan tidak adanya cetak ulang buku tersebut diatas.

Maka bersama ini **saya mohon izin untuk dapat mempublish draft tulisan buku tersebut di Internet secara bebas (gratis)**. Hal ini mengingat adanya sejumlah pertanyaan/permintaan buku tersebut kepada saya secara pribadi (karena sudah tidak ditemukannya lagi buku tersebut secara on line di toko2 internet) dan disisi lain saya juga ingin draft tulisan pada buku tersebut bermanfaat bagi pihak yang membutuhkannya (terutama mahasiswa).

Terimakasih atas perhatiannya dan saya menunggu tanggapannya.

Salam  
Iwan Setiawan

## KATA PENGANTAR

Tak dapat dipungkiri, sampai saat ini kontrol **PID** (**Proporsional Integral Derivative**) merupakan satu-satunya strategi yang paling banyak diadopsi pada pengontrolan proses industri. Berdasarkan survey, 97% industri yang bergerak dalam bidang proses (seperti industri kimia, *pulp*, makanan, minyak dan gas) menggunakan PID sebagai komponen utama dalam pengontrolannya (sumber: Honeywell, 2000).

Kepopuleran PID sebagai komponen kontrol proses dilatarbelakangi terutama oleh kesederhanaan struktur, serta kemudahan dalam melakukan tuning parameter kontrolnya. Pada tingkat pengoperasian, seorang operator tidak dituntut untuk menguasai pengetahuan matematika yang relative rumit, melainkan hanya dibutuhkan pengalaman lapangan serta sedikit pengetahuan teori kontrol dasar saja.

Selain sederhana, kepopuleran PID disebabkan juga oleh alasan histories. Dalam hal ini, PID telah diterapkan di industri secara luas jauh sebelum era digital berkembang, yaitu dimulai sekitar tahun 1930-an, dimana saat itu strategi kontrol PID diimplementasikan dengan menggunakan rangkaian elektronika analog, bahkan banyak diantaranya direalisasikan dengan menggunakan komponen mekanis dan pneumatis murni.

Seiring dengan perkembangan teknologi digital dan *solid state*, dewasa ini produk PID komersil muncul di pasaran dalam beragam model dan bentuk, yaitu dari sekedar modul jenis *special purpose process controller* (seperti *Temperature Controller*, *Pressure Controller*, dan sebagainya) sampai modul kontrol jenis *general purpose process controller* atau yang lebih dikenal dengan nama DCS (*Distributed Control System*). Bahkan perkembangan terakhir, modul PID ini juga umum dijumpai dalam bentuk modul independen pada sistem PLC (*Programmable Logic Controller*)

Walaupun peranannya begitu besar dalam menentukan keberhasilan operasi proses industri, tetapi patut disayangkan sampai saat ini sangat jarang buku berbahasa Indonesia yang secara khusus membahas kontrol PID dan implementasinya pada kontrol proses industri.

Terkait dengan hal tersebut, maka penulis terdorong untuk memberi sedikit kontribusi dengan menulis buku ini. Seperti yang tersirat dalam judul buku, materi yang dibahas dalam buku ini pada dasarnya berhubungan dengan implementasi kontrol PID untuk proses industri.

Sebagai sebuah buku teks kontrol, Materi pada buku ini dirancang se-sistematis mungkin sedemikian sehingga pembaca dapat mempelajarinya secara mandiri tanpa perlu bimbingan dari dosen atau instruktur. Sasaran pembaca yang dituju adalah mahasiswa Teknik (terutama mahasiswa Teknik Kimia, Fisika dan Elektro) serta para praktisi dan personel lapangan yang bekerja dalam bidang industri proses.

Tanpa mengabaikan aspek teori yang dirasa penting dan relevan dengan materi utama, buku ini lebih menitikberatkan segi praktis yang dapat dijumpai di lapangan. Bagi pembaca (terutama mahasiswa) yang tidak dapat mengakses pengontrolan sebuah proses nyata secara langsung, pembaca disarankan menggunakan perangkat simulasi komputer untuk mempraktekan materi yang disajikan dalam buku ini.

Buku ini terdiri dari lima bab dan dua buah lampiran penting yang disusun secara runtun.

Pada **Bab 1**, terlebih dulu akan dibahas dua jenis model proses yang umum dijumpai di industri : model *Self Regulating Process* dan *Non Self Regulating Process*. Pembahasan kedua jenis proses tersebut diberikan dalam bentuk kasus-kasus nyata yang dapat ditemukan di lapangan serta hasil simulasi model matematisnya dengan Simulink. Materi ini terlebih dulu disajikan mengingat pengetahuan model proses merupakan salah satu kunci keberhasilan dalam melakukan tuning parameter PID yang tepat untuk proses yang akan atau sedang dikontrol.

Berbekal pengetahuan model proses, dalam **Bab 2** akan dibahas berbagai struktur atau algoritma PID untuk proses industri. Agar pembaca mendapat gambaran luas tentang jenis dan tipe PID yang digunakan di lapangan, struktur PID yang dibahas dalam bab ini secara umum mengacu pada struktur PID yang dapat dijumpai dalam modul-modul PID komersil.

Selain membahas struktur kontrol PID, pada Bab 2 disajikan pula berbagai metode tuning empiris berdasarkan percobaan sederhana yang dikenal dengan nama *Bump Test Experiment*.

Disamping kemampuan-kemampuan standar, modul PID komersil dewasa ini umumnya telah dilengkapi juga dengan berbagai fitur pendukung seperti fasilitas Autotuning dan Gain Scheduling. Untuk kedua kemampuan tersebut, prinsip kerjanya akan dibahas secara detail pada **Bab 3**.

Materi lanjutan dari kontrol proses seperti skema *Cascade Control*, *feedforward-feedback control*, dan *Ratio control* akan disajikan pada **Bab 4**. Skema-skema kontrol yang sifatnya *multiloop* tersebut digunakan terutama untuk pengontrolan proses yang memiliki parameter dinamis yang sangat dominan.

Karena modul PID komersil yang beredar sekarang ini didominasi oleh modul digital, maka penting bagi pembaca untuk mengetahui struktur PID dalam bentuk digital-nya. Materi tentang PID digital tersebut dapat dijumpai pada **Bab 5** yang juga merupakan bab terakhir dari buku ini.

Bagi pembaca yang berniat merealisasikan skema kontrol PID dalam sistem komputer atau sistem microcontroller secara *real time*, bab 5 juga dapat dijadikan rujukan karena dalam bab ini dibahas realisasi PID digital dalam bentuk algoritma komputer dengan menggunakan bahasa C sebagai bahasa pemrogramannya ( Bahasa C dipilih mengingat dewasa ini banyak microcontroller telah mendukung penggunaan bahasa tersebut).

Agar mendapatkan hasil yang optimal dari buku ini, pembaca diharapkan memiliki dasar pengetahuan tentang teori kalkulus dan konsep fungsi alih (khusus yang belum pernah mendapatkan materi fungsi alih, pembaca secara cepat dapat merujuk pada lampiran A).

Karena proses industri serta kontrol PID itu sendiri memiliki model matematis dinamis yang secara intuitive perilakunya relative sukar dipahami. Maka pembaca disarankan menggunakan perangkat lunak komputer untuk mensimulasikan skema-skema kontrol yang dibahas dalam keseluruhan buku ini. Salah satu perangkat lunak yang mudah serta telah luas penggunaannya adalah Matlab Simulink (Pengenalan Matlab Simulink dapat ditemukan pada lampiran B).

Walaupun materi pada buku ini telah dipersiapkan secermat mungkin, tetapi seperti kata pepatah tiada gading yang tidak retak, penulis menyadari masih banyak kekurangan dalam buku ini. Oleh karena itu untuk kesempurnaan buku ini dimasa datang, penulis mengharap masukan dan kritik dari pembaca. Kritik atau saran dapat pembaca tujukan pada alamat –email : setiaone.iwan@gmail.com.

Sebagai penutup, dalam kesempatan ini penulis ingin mengucapkan banyak terimakasih kepada pimpinan teknik elektro UNDIP, Bapak Ir. Sudjadi MT dan Bapak Ir. Sulasno atas segala dukungan dan motivasinya. Ucapan terimakasih dan penghargaan juga penulis sampaikan kepada Bapak DR. Lilik Soetiarso dan Ir. Atris Suyanto Hadi, MT atas kerjasama yang dilakukan selama ini, Bapak Jasmadi dan Penerbit yang telah memfasilitasi dan bersedia menerbitkan buku ini, serta rekan-rekan penulis di Teknik Elektro UNDIP.

Semarang, Mei 2008

Penulis

# Daftar Isi

## Kata Pengantar

## Daftar Isi

### 1. Dinamika Dan Model Kontrol Proses Industri

- 1.1. Pendahuluan
- 1.2. Sekilas tentang Model Proses dan Pengontrolan
- 1.3. Sekilas tentang Diagram Instrumentasi Proses (P&ID)
- 1.4. Model Self Regulating Process
  - 1.4.1. Representasi Matematis Model Proses FOPDT dengan Fungsi Alih
  - 1.4.2. Akomodasi Model Gangguan pada Proses FOPDT
- 1.5. Model Non Self Regulating Process
- 1.6. Contoh Soal dan Penyelesaian
- 1.7. Ringkasan
- 1.8. Soal Latihan

### 2. Kontrol PID untuk Proses Industri: Berbagai Struktur dan Metode Tuning Praktis

- 2.1. Pendahuluan
- 2.2. Berbagai Struktur Kontrol PID dan Metode Tuningnya
  - 2.2.1. Struktur PID Ideal
  - 2.2.2. Struktur PID Tipe B
  - 2.2.3. Struktur PID Tipe C
  - 2.2.4. Struktur PID dengan Filter (PIDF)
  - 2.2.5. Struktur PID Standar ISA dan PID bentuk umum
  - 2.2.6. Struktur PID Serial
- 2.3. Struktur Kontrol PID dengan Kompensator Delay: Smith Predictore dan Predictive Proporsional Integral (pPI)
- 2.4. Dua Fitur Penting pada Sistem Kontrol PID Komersil: AntiWindUp dan Bumpless Transfer
  - 2.4.1. Phenomena *WindUp* pada kontrol PID dan Antisipasinya
  - 2.4.2. BumpLess Transfer
- 2.5. Contoh Soal dan Penyelesaian
- 2.6. Ringkasan
- 2.7. Soal Latihan



### **3. Autotuning dan Gain Scheduling**

#### 3.1. Pendahuluan

#### 3.2. Autotuning Metode Respon Transien

#### 3.3. Autotuning Metode Umpan Balik Relay

##### 3.3.1. Konsep Dasar Kontrol umpan balik Relay

##### 3.3.2. Perhitungan parameter kontrol PID berdasarkan informasi magnitude dan perioda osilasi output proses

##### 3.3.3. Mekanisme Autotuning Metode umpan balik relay

#### 3.4. Gain Scheduling

##### 3.4.1. Studi Kasus Gain Scheduling pada Sistem Heat Exchanger

##### 3.4.2. Studi Kasus Gain Scheduling pada Kontrol Level dengan Penampang Tanki Non Linear

#### 3.5. Contoh Soal dan Penyelesaian

#### 3.6. Ringkasan

#### 3.7. Soal Latihan

### **4. Skema Kontrol PID Lanjut: Cascade, Feedforward- Feedback Dan Ratio Control**

#### 4.1. Pendahuluan

#### 4.2. Skema Cascade Control

#### 4.3. Skema Feedforward-feedback control

#### 4.4. Skema Ratio Control

#### 4.5. Contoh Soal dan Penyelesaian

#### 4.6. Ringkasan

#### 4.7. Soal latihan

### **5. Kontrol PID Digital dan Realisasi Program**

#### 5.1. Pendahuluan

#### 5.2. Persamaan PID Digital dan Realisasi Program

#### 5.3. Anti Wind Up dan Realisasi Program

#### 5.4. Bumpless transfer dan Realisasi Program

#### 5.5. Sistem Kontrol Umpan Balik Relay dan Realisasi Program

#### 5.6. Contoh Soal dan Penyelesaian

#### 5.7. Ringkasan

#### 5.8. Soal Latihan

### **Lampiran A: Pengantar Transformasi Laplace**

### **Lampiran B. Pengenalan Matlab dan Simulink**

# Bab 1

## DINAMIKA DAN MODEL KONTROL PROSES INDUSTRI

### Tujuan Bab

Dalam bab ini terlebih dulu akan ditinjau dinamika dan model kontrol proses yang umum dijumpai di Industri. Setelah menyelesaikan bab ini, anda diharapkan dapat:

- Menjelaskan pengertian praktis model proses serta kaitannya dengan masalah pengontrolan
- Menggambarkan kontrol proses industri dalam bentuk diagram blok maupun diagram instrumentasi proses
- Menjelaskan pengertian proses *Self Regulating* dan proses *Non Self Regulating* yang umum dijumpai di industri
- Melakukan identifikasi parameter model proses *Self regulating* dan *non Self regulating*
- Merepresentasikan model proses secara matematis dalam bentuk fungsi alih
- Menurunkan model matematis proses berdasarkan Persamaan Fisika
- Melakukan simulasi dinamika proses dengan menggunakan Simulink

### Bacaan Prasyarat

- Lampiran A. Pengantar Transformasi Laplace
- Lampiran B. Pengenalan Matlab dan Simulink

### 1.1 Pendahuluan

Pada level terendah, salah satu permasalahan yang dihadapi oleh industri proses (seperti industri kimia, makanan, pulp, pupuk, minyak dan gas) adalah regulasi atau pengontrolan berbagai macam variabel proses pada titik-titik kerja yang diharapkan. Beberapa variabel yang umum dilakukan tindakan pengontrolan tersebut meliputi temperature fluida (atau gas) pada sistem *heat exchanger*, tekanan pada reaktor, laju fluida pada pipa, level pada tanki penampung, konsentrasi bahan kimia dan sebagainya.

Terkait dengan masalah pengontrolan, salah satu kunci utama keberhasilannya adalah pengetahuan mengenai karakteristik dinamik atau model prosesnya itu sendiri. Pengetahuan model sangat penting mengingat secara teknis terdapat hubungan antara proses yang akan dikontrol dengan parameter kontroler (PID) yang harus di-*tuning*. Dalam hal ini, parameter PID optimal pada dasarnya dapat dicari secara lebih pasti (tanpa coba-coba) berdasarkan model dan nilai parameter proses yang diketahui.

Tidak tergantung dari variabel proses yang dikontrol (apakah temperature, tekanan, level, aliran, konsentrasi dan sebagainya), kontrol proses industri dalam banyak kasus umumnya akan menunjukkan salah satu karakteristik dari dua model matematis berikut:

- **Model proses self regulating:** Model proses yang bersifat stabil. Untuk kepentingan perancangan dan tuning parameter kontrol PID-nya, model proses

ini dapat didekati oleh sebuah model matematis yang dikenal dengan nama model **FOPDT** (*First Order Plus Dead Time*) yang hanya dicirikan oleh tiga buah parameter: Keterlambatan transportasi ( $L$ ), Konstanta waktu proses ( $T$ ) dan Gain statis proses ( $K$ ).

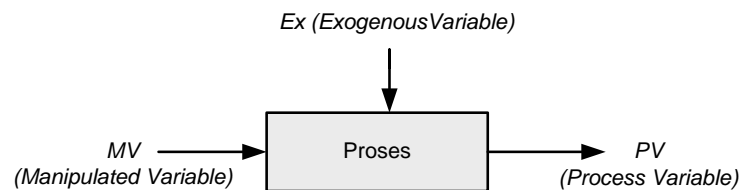
- **Model proses non self regulating:** Model proses yang tidak stabil. Salah satu yang sering dijumpai di industri adalah model **IPDT** (*Integrating Plus Dead Time*) yang hanya dicirikan oleh dua buah parameter saja: Keterlambatan transportasi ( $L$ ) dan Gain integratif proses ( $K^*$ ).

Untuk kedua model proses tersebut, parameter kontrol PID yang optimal dapat dicari berdasarkan nilai parameter-parameter proses asosiasinya.

Dalam banyak kasus pengontrolan, pengetahuan parameter proses itu sendiri secara praktis umumnya dicari lewat eksperimen. Jika dilaksanakan secara benar, eksperimen ini dapat dilangsungkan bahkan pada kontrol proses yang sedang berjalan tanpa menyebabkan operasi pengontrolan mengalami *upset* terlalu parah dari kondisi normalnya.

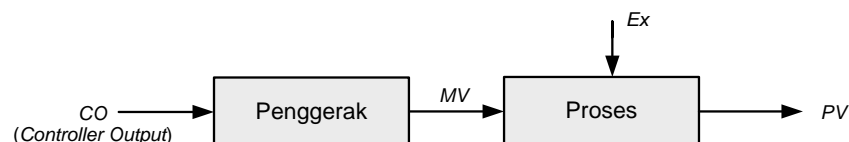
## 1.2. Sekilas Tentang Model Proses dan Pengontrolan

Model proses pada dasarnya adalah hubungan sebab akibat antara variabel input dengan variabel output proses (*Process Variable-PV*). Ditinjau dari segi pengontrolan, variabel input proses itu sendiri dapat dibedakan menjadi dua jenis: (1) variabel input yang dapat dimanipulasi (*Manipulated Variable-MV*) dan (2) variabel input exogenous (*Exogenous Variable-Ex*) yang umumnya tidak dapat dimanipulasi secara langsung. Gambar 1.1 berikut memperlihatkan diagram blok sebuah model proses yang disederhanakan.



Gambar 1.1 Diagram blok model proses

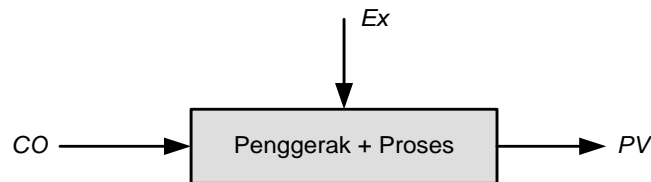
- **Variabel input yang dapat dimanipulasi (*Manipulated Variable - MV*) :** Variabel input yang besarnya secara langsung dapat dimanipulasi oleh kontroler. Tergantung dari proses yang dikontrol, variabel tersebut dapat saja berupa laju aliran fluida, laju aliran uap, laju aliran panas, laju konsentrasi bahan kimia, dan sebagainya. Secara teknis, manipulasi variabel oleh kontroler umumnya dilakukan lewat elemen penggerak yang terhubung dengan proses (lihat gambar 1.2). Beberapa elemen penggerak yang sering dijumpai dalam kontrol proses industri diantaranya adalah *control valve*, *heater*, kompresor dan pompa.



Gambar 1.2 Manipulasi *MV* dilakukan oleh kontroler lewat sebuah penggerak

Ditinjau dari kepentingan pengontrolan praktis, model kontrol proses (hubungan input-output) yang perlu diketahui oleh perancang atau operator lapangan adalah model yang merelasikan variabel output proses yang dikontrol-*PV* dengan

variabel input proses yang berasal dari keluaran kontroler- $CO$  (dengan kata lain, penggerak yang terhubung pada proses dapat dipandang sebagai bagian integral dari proses tersebut – lihat gambar 1.3).

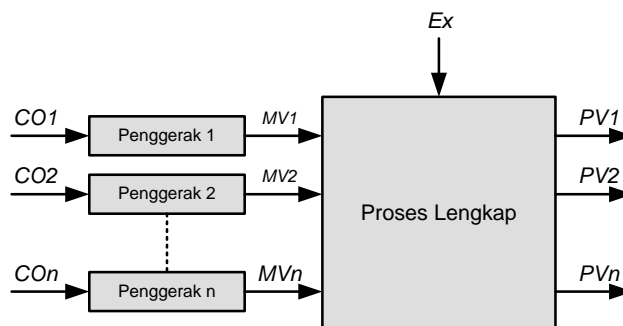


**Gambar 1.3** Penggerak dapat dianggap sebagai bagian tak terpisahkan dari proses

- **Variabel input Exogenous (*Exogenous Variable-Ex*):** Variabel input yang tidak dapat dimanipulasi secara langsung oleh kontroler. Jika input *exogenous-Ex* suatu saat mengalami deviasi atau perubahan disekitar nilai *steady*-nya, maka perubahan tersebut pada dasarnya berlaku sebagai gangguan (*disturbance*).

Dalam sebuah sistem kontrol proses, variabel input *exogenous* ini dapat muncul dari berbagai macam sumber. Sebagai contoh, pada pengontrolan temperature ruang atau reaktor, salah satu variabel *exogenous* yang mempengaruhi temperature ruang atau reaktor tersebut adalah temperature lingkungan. Perhatikan, dalam hal ini yang berlaku sebagai gangguan pada sistem pengontrolan diatas bukanlah nilai temperature lingkungannya itu sendiri, melainkan variasi atau perubahan temperature lingkungan disekitar nilai *ambient*-nya: Tidak peduli berapa besar nilai temperature nominal lingkungannya, selama temperature tersebut tidak berubah, maka aksi pengontrolan pada dasarnya tidak akan tertanggu.

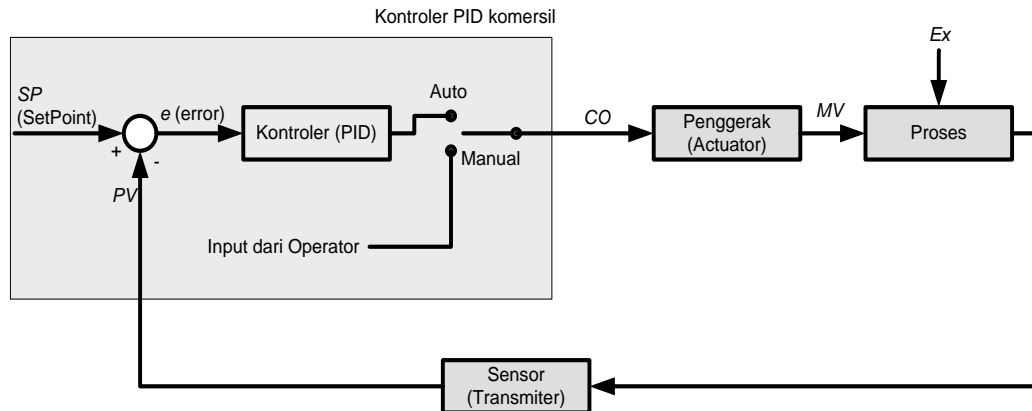
Walaupun dalam sebuah sistem kontrol proses yang lengkap, jumlah masing-masing variabel input (yang dapat dimanipulasi) dan variable output proses dimungkinkan lebih dari satu jenis (proses MIMO: *Multiple Input Multiple Output*), tetapi dalam pengontrolannya, sampai saat ini pengontrolan SISO (*Single Input Single Output*) adalah prinsip pengontrolan yang umum dijumpai di industri. Dalam pengontrolan ini, setiap variabel output proses hanya terasosiasi dengan satu sumber sinyal  $CO$  dan satu penggerak saja (lihat gambar 1.4).



**Gambar 1.4** Prinsip pengontrolan SISO pada proses industri

Penggunaan prinsip pengontrolan SISO tersebut pada dasarnya dilatarbelakangi terutama karena kesederhanaan perancangan dan tuning sistem kontrolnya. Gambar 1.5 berikut memperlihatkan diagram blok pengontrolan satu buah variabel proses dengan menggunakan modul kontroler PID komersil (misal berupa perangkat DCS: *Distributed*

*Control System* atau PLC: *Programmable Logic Controller*) yang umum di jumpai di industri.



**Gambar 1.5.** Diagram blok pengontrolan proses dengan menggunakan kontroler komersil

#### Catatan Penting tentang Mode Kontrol pada Modul PID komersil

Modul PID komersil yang beredar di pasaran umumnya menyediakan minimal dua mode pengontrolan kontinyu yang dapat digunakan secara bebas: Mode Manual dan Mode Auto (lihat gambar 1.5).

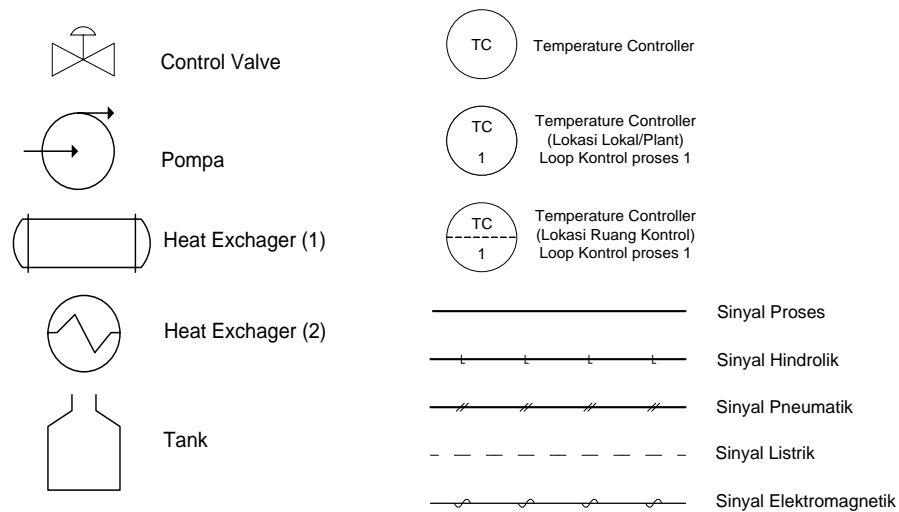
Pengontrolan mode manual adalah pengontrolan dengan menggunakan prinsip *open loop* atau loop terbuka. Dalam mode ini, besarnya output Kontroler – CO diatur oleh operator. Mode manual digunakan terutama pada *Set up* awal proses kontrol, atau jika pengontrolan secara *closed loop* menghasilkan unjuk kerja pengontrolan yang tidak memuaskan. Menurut sebuah survey, hampir 30% kontrol proses industri menggunakan mode manual dalam operasi kontrolnya [Honeywell, 2000].

Pengontrolan mode Auto adalah pengontrolan dengan menggunakan prinsip *closed loop* atau umpan balik. Dalam mode ini, kontroler bekerja berdasarkan error (selisih antara SetPoint – SP dengan Output Proses – PV). Unjukkerja pengontrolan pada mode Auto sangat ditentukan oleh nilai parameter PID yang di-*setting* pada sistem kontrol tersebut.

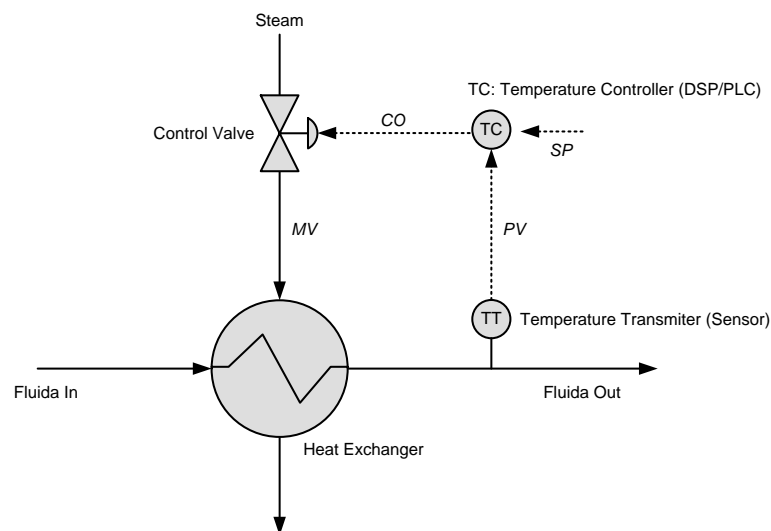
### 1.3. Sekilas Tentang Diagram Instrumentasi Proses (P&ID)

Khusus dalam bidang kontrol proses, selain diagram blok (seperti gambar 1.5), diagram lain yang sering digunakan untuk merepresentasikan pengontrolan variabel proses di industri adalah diagram instrumentasi proses atau lebih dikenal dengan nama *Piping and Instrumentation Diagram (P&ID)*. Diagram Instrumentasi proses ini secara visual memperlihatkan beragam simbol perangkat keras dan simbol interkoneksi pada proses yang dikontrol secara lengkap (lihat gambar 1.6).

Berbeda dengan diagram blok yang hanya menggambarkan aliran sinyal informasi kontrol saja, penggambaran kontrol proses dengan menggunakan representasi P&ID ini secara langsung akan memperlihatkan berbagai sumber daya yang digunakan dalam proses tersebut. Sebagai contoh, jika variabel proses yang dikontrol pada gambar 1.5 adalah temperature fluida output pada sebuah sistem *heat exchanger* (misal dengan sumber energi panas berupa uap - *steam*), maka representasi diagram instrumentasi proses-nya akan nampak seperti pada gambar 1.7.



**Gambar 1.6.** Beberapa simbol perangkat keras dan interkoneksi yang umum dijumpai pada P&ID



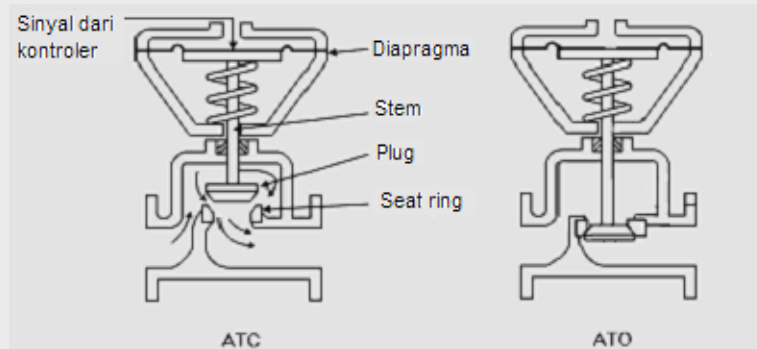
**Gambar 1.7.** Diagram instrumentasi proses sistem pengontrolan temperature pada sistem *heat exchanger*

Seperti terlihat pada gambar 1.7 diatas, pengontrolan temperature fluida output (*PV*) dalam hal ini dilakukan dengan cara mengontrol bukaan *control valve* (sebagai penggerak) yang terhubung dengan sistem *heat exchanger*. Tergantung dari jenis *control valve* yang digunakan (lihat catatan penting tentang *control valve*), semakin besar sinyal kontrol *CO* yang berasal dari modul *Temperature Controller* – TC (dapat berupa DCS/PLC dengan mode manual atau Auto), aliran uap yang dilewatkan dapat semakin besar (*valve* semakin terbuka) atau semakin kecil (*valve* semakin tertutup). Besar kecilnya aliran uap secara langsung tentunya akan mempengaruhi temperature fluida pada sisi outlet *heat exchanger* tersebut.

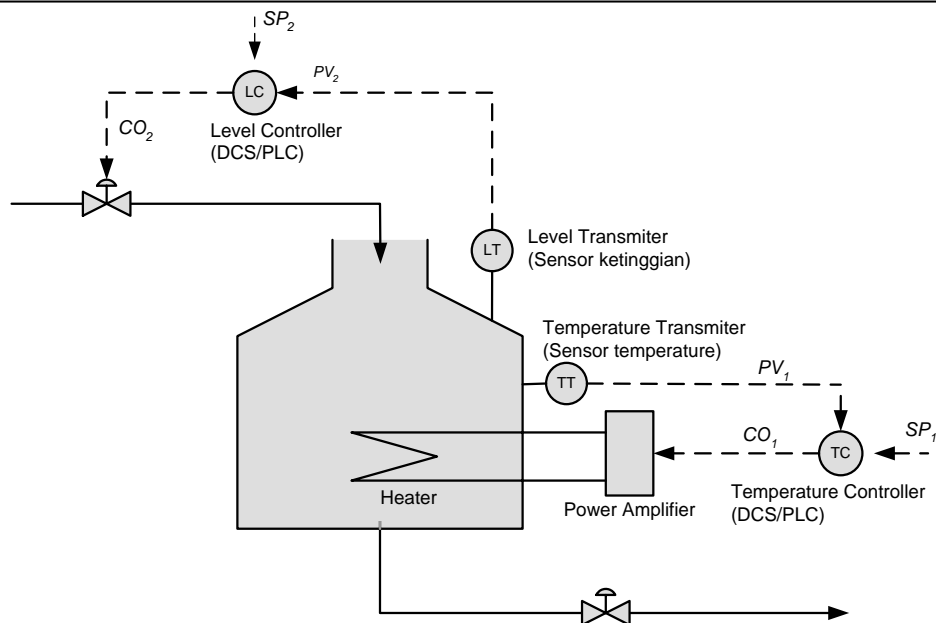
### Catatan penting tentang Control Valve

Berdasarkan prinsip kerjanya ada dua jenis *control valve* yang umum digunakan di industri proses:

1. **Control Valve jenis ATC (*Air To Close*):** Semakin besar tekanan udara penggerak atau sinyal kontrol, bukaan *valve* akan semakin mengecil. Tipe ini dikenal juga dengan nama *fail open*
2. **Control Valve jenis ATO (*Air To Open*):** Semakin besar tekanan udara penggerak atau sinyal kontrol, bukaan *valve* akan semakin membesar. Tipe ini dikenal juga dengan nama *fail close*



**Gambar 1.8.** Prinsip kerja Control Valve berdasarkan tipe penggeraknya



**Gambar 1.9.** Diagram instrumentasi proses untuk kasus pengontrolan level dan temperature fluida pada sebuah tanki

Selain memperlihatkan beragam sumber daya perangkat keras, representasi kontrol proses dengan menggunakan P&ID pada dasarnya dapat menggambarkan keseluruhan kontrol proses secara lengkap, Gambar 1.9 memperlihatkan contoh lain P&ID untuk kasus

pengontrolan dua buah variabel proses (level dan temperature fluida) pada sebuah tanki atau reaktor.

**Catatan penting tentang Modul kontroler dalam sebuah P&ID**

Walaupun dalam sebuah diagram instrumentasi proses terdapat beberapa simbol modul kontroler yang terpisah atau berbeda (misal dalam gambar 1.9 ada dua simbol kontroler: TC dan LC) tapi dalam prakteknya, simbol-simbol tersebut bisa saja mengacu pada sebuah perangkat keras kontroler DCS/PLC yang sama (Perangkat DCS/PLC umumnya menyediakan lebih dari satu buah *loop* pengontrolan )

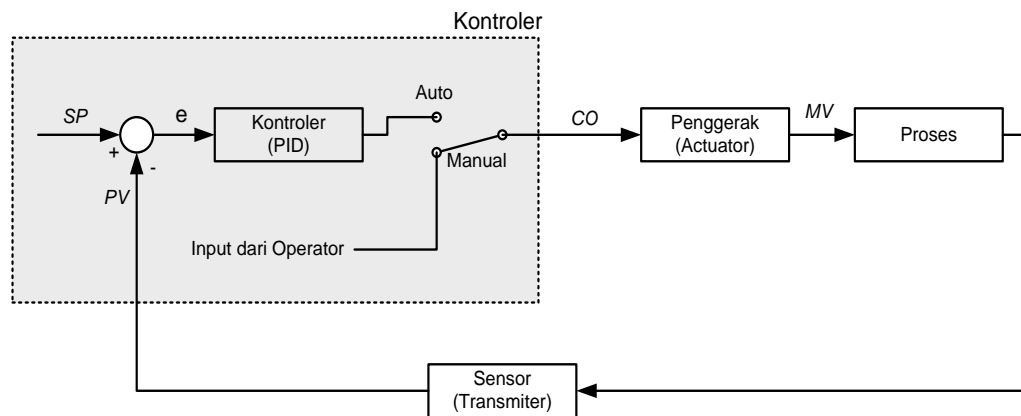
#### 1.4. Model Self Regulating Process

Model *self regulating process* adalah model yang paling umum dijumpai di industri proses. Model ini hampir muncul pada setiap kasus pengontrolan praktis, yaitu seperti pengontrolan temperature pada sistem *heat exchanger*, pengontrolan level fluida pada sistem tanki penampung dan sebagainya.

Seperti telah sedikit disinggung pada **subbab Pendahuluan**, model *self regulating process* pada dasarnya dapat didekati oleh sebuah model matematis **FOPDT** (First Order Plus Dead Time) yang hanya dicirikan oleh tiga buah parameter berikut:

1. *Process transport delay* (delay atau keterlambatan transportasi proses) -  $L$
2. *Process time constant* (konstanta waktu proses) -  $T$
3. *Process static gain* (gain statis proses) -  $K$

Ketiga parameter yang menggambarkan dinamika proses tersebut, secara praktis dapat diperoleh atau diidentifikasi melalui eksperimen sederhana *Bump test* atau test sinyal tangga secara *open loop* pada mode kontrol manual (lihat gambar 1.10).



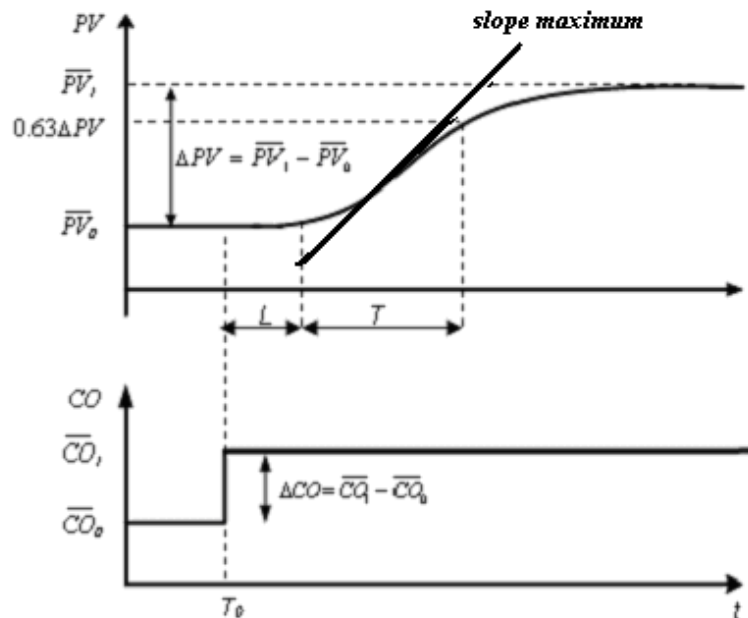
**Gambar 1.10.** Eksperimen bump test pada mode kontrol manual (loop terbuka)

Secara teknis, eksperimen *bump test* dilakukan dengan cara memberi perubahan tangga (*step*) sinyal output kontroler –  $CO$  oleh operator pada saat proses telah mengalami keadaan *steady* (menetap) disekitar titik kerja nominalnya. Respon variabel output ( $PV$ ) kemudian direkam dan dianalisis dengan menggunakan perangkat lunak tertentu atau dapat juga dianalisis secara manual oleh operator yang bertanggung jawab terhadap proses tersebut.

Gambar 1.11 berikut ini adalah grafik tipikal hubungan sebab akibat antara perubahan sinyal output kontroler (sebagai input proses) dengan variabel output proses hasil dari eksperimen *bump test* pada keadaan *steady* tertentu untuk model proses FOPDT (Secara



praktis, grafik tersebut dapat saja merupakan hasil uji *bump test* pada sistem *heat exchanger*- gambar 1.7, dimana dalam hal ini penggerak yang digunakan adalah *control valve* jenis ATO- Air To Open).



**Gambar 1.11.** Respon tangga pada eksperimen *bump test* untuk model FOPDT

Dengan mengacu grafik respon tangga pada gambar 1.11, parameter-parameter proses FOPDT dapat dicari/dihitung sebagai berikut:

- **Process Transport delay (keterlambatan transportasi proses) -  $L$**  : Waktu keterlambatan transportasi atau waktu ketidakpastian yang terjadi pada proses dihitung sejak terjadi perubahan tangga pada  $CO$  sampai variabel proses ( $PV$ ) yang dikontrol mulai menanggapi perubahan input  $CO$  tersebut. Tergantung variabel prosesnya, rentang waktu keterlambatan yang terjadi dapat berkisar dari satuan detik sampai menit. Dalam bidang kontrol proses, keterlambatan transportasi atau waktu mati yang relative lama (dibandingkan dengan konstanta waktu proses) seringkali menimbulkan permasalahan unjukkerja pengontrolan yang relative sukar diatasi dengan algoritma kontrol PID biasa.

Secara praktis, penyebab utama atau yang paling mendominasi terjadinya keterlambatan transportasi ini adalah posisi penggerak dari *plant* proses yang sedang dikontrol. Misal dalam sistem *heat exchanger* (gambar 1.7), keterlambatan transportasi yang terjadi sangat ditentukan oleh jarak *control valve* dari *plant heat exchanger* tempat berlangsungnya proses pertukaran panas tersebut: Semakin jauh posisi *control valve* dari *plant*, keterlambatan transportasi akan semakin lama.

- **Process Time Constant (Konstanta waktu proses) -  $T$**  : waktu yang diperlukan sehingga nilai  $PV$  mencapai kurang lebih 63% dari keadaan *steady* akhirnya. Perhitungan nilai konstanta waktu dimulai sejak  $PV$  mulai menanggapi perubahan  $CO$  (setelah waktu tunda berlalu). Seperti halnya keterlambatan transportasi, konstanta waktu proses dapat berkisar dari satuan detik sampai menit.

Selain dengan cara mengamati respon dari grafik, Konstanta waktu proses dapat juga dihitung berdasarkan gradien atau slope maksimum yang terjadi pada saat transien.

Dengan menggunakan teknik ini, konstanta waktu dapat dicari berdasarkan perumusan (1.1) dibawah :

$$T = \frac{\Delta PV}{slope \max} \quad (1.1)$$

Dalam hal ini  $\Delta PV$  dan  $slope \max$  berturut-turut adalah deviasi dan gradien output proses maksimum pada saat transien (lihat gambar 1.11).

Besar kecilnya konstanta waktu tersebut pada dasarnya menunjukkan kecepatan respon proses: semakin kecil nilai konstanta waktu, maka respon proses semakin cepat. Nilai ini pada dasarnya ditentukan oleh tiga hal utama berikut:

- Dimensi *plant* tempat kontrol proses berlangsung
- Jenis material serta beban yang terlibat pada kontrol proses
- Kekuatan atau daya penggerak

Misal untuk sistem *heat exchanger* pada gambar 1.7, konstanta waktu proses ditentukan secara langsung oleh ukuran *heat exchanger*, jenis serta debit fluida yang dikontrol dan juga tekanan nominal dari *steam* pemanas. Sebagai contoh, respon proses pada gambar 1.7 akan semakin cepat (atau konstanta waktu- $T$  akan semakin kecil) jika dimensi *heat exchanger* berukuran kecil, aliran fluida yang dikontrol relative cepat serta tekanan *steam* yang digunakan pada proses pemanasan semakin tinggi.

Ditinjau dari sudut pengontrolan umpan balik, perbandingan antara waktu mati dengan konstanta waktu proses ( $L/T$ ) akan sangat menentukan perlu tidaknya penggunaan kompensator atau prediktor tambahan pada kontroler konvensional (PID): Jika  $L/T > 1$ , maka kontroler umumnya memerlukan kompensator tambahan agar unjuk kerja kontroler relatif memuaskan.

Nilai konstanta waktu proses ini secara langsung juga mempengaruhi waktu cuplik ( $T_c$ ) yang harus digunakan/dipilih pada sebuah kontroler digital komersil (sistem DCS atau PLC). Agar tidak terjadi *aliasing*, waktu cuplik yang digunakan atau dipilih pada kontroler digital harus jauh lebih kecil dari konstanta waktu proses yang akan dikontrol (nilai  $T_c$  yang dipilih umumnya paling tidak sebesar  $0.1T$ )

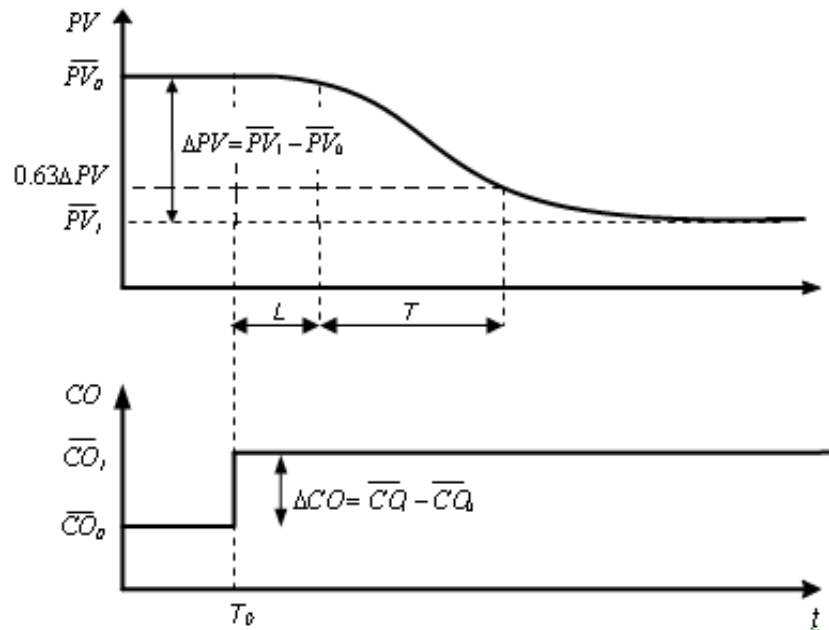
- **Process static gain (gain statis proses) –  $K$**  : Gain statis proses adalah perbandingan perubahan  $PV$  terhadap perubahan  $CO$  dalam keadaan *steady*-nya :

$$K = \frac{\Delta PV}{\Delta CO} = \frac{\overline{PV_1} - \overline{PV_0}}{\overline{CO_1} - \overline{CO_0}} \quad (1.2)$$

Nilai gain proses ini secara langsung menunjukkan kesensitifan dari proses, semakin besar gain statis maka proses semakin sensitif: perubahan kecil pada  $CO$  akan menghasilkan deviasi  $PV$  yang relatif besar (lihat relasi pada persamaan 1.2). Seperti halnya parameter konstanta waktu, gain proses besarnya sangat ditentukan oleh dimensi *plant*, daya penggerak serta beban yang terlibat pada proses tersebut.

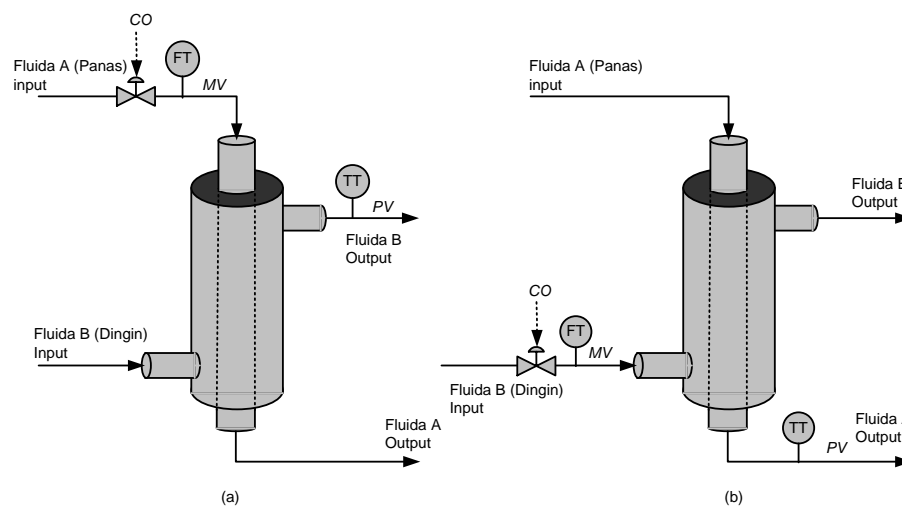
Tergantung dari proses dan sifat penggeraknya, gain statis proses pada dasarnya dapat bernilai positive atau negative (hal ini berbeda dengan dua parameter proses sebelumnya, yaitu  $L$  dan  $T$  yang selalu bernilai positif). Sebagai contoh, jika *control valve* yang digunakan pada sistem *heat exchanger* - gambar 1.7 bertipe *fail open* atau *ATC : Air To Close*, maka perubahan nilai tangga positive pada  $CO$  akan

menyebabkan bukaan *valve* semakin mengecil (semakin menutup), sehingga tentunya nilai temperature output - *PV* juga akan semakin menurun atau menuju nilai *steady* yang lebih negatif dari nilai sebelumnya (lihat gambar 1.12).



**Gambar 1.12.** Respon tangga untuk gain statis proses bernilai negative

Secara praktis, tanda nilai gain statis proses tersebut dapat dipandang sebagai arah pergerakan variabel output proses -- *PV* relatif terhadap perubahan nilai input penggerak proses (output controller -- *CO*): Jika *K* bernilai positive maka pengaruh perubahan *CO* terhadap *PV* akan bersifat searah (*direct*), sedangkan jika *K* bernilai negative, maka pengaruh perubahan *CO* terhadap *PV* akan bersifat kebalikannya (*reverse*). Untuk lebih memahami penjelasan tersebut, lihat ilustrasi sistem pemanasan dan pendinginan suatu fluida pada gambar 1.13 dibawah.



**Gambar 1.13.** Proses Pemanasan (a) dan proses pendinginan (b) fluida

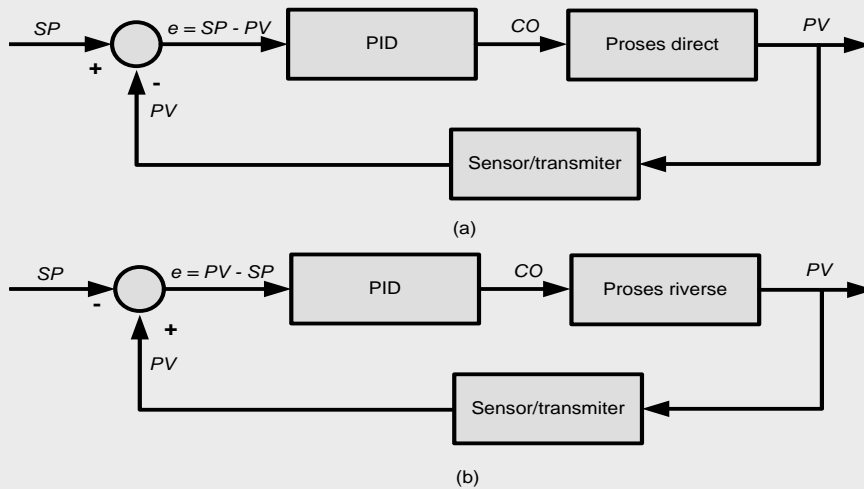
Dengan menganggap kedua jenis *control valve* yang digunakan bertipe *ATO* (*Air To Open*), maka dalam proses pemanasan (lihat gambar 1.13(a)), semakin besar sinyal

control ( $CO$ ) yang diberikan, debit aliran fluida A (panas) akan semakin besar, hal ini tentunya akan menyebabkan temperature keluaran fluida B (variabel yang dikontrol) akan semakin tinggi juga, atau dengan kata lain perubahan  $PV$  terhadap perubahan  $CO$  bersifat *direct* ( $K$  bernilai positive). Kasus sebaliknya terjadi pada proses pendinginan (gambar 1.13(b)), semakin besar sinyal kontrol ( $CO$ ) yang diberikan maka debit aliran fluida B (dingin) akan semakin besar dan hal ini tentunya akan menyebabkan temperature keluaran fluida A (variabel yang dikontrol) akan semakin menurun (menjadi lebih dingin), atau dengan kata lain perubahan  $PV$  terhadap perubahan  $CO$  bersifat *reverse* ( $K$  bernilai negative).

**Catatan Penting tentang hubungan tanda gain proses statis dengan Mode aksi kontrol pada Modul PID komersil yang harus dipilih**

Untuk mengakomodasi perbedaan tanda dari gain statis proses yang dikontrol, Modul PID komersil umumnya menyediakan dua mode aksi *loop* tertutup: Mode aksi *Reverse* dan mode aksi *Direct* yang pilihannya harus disesuaikan dengan sifat proses yang akan dikontrol. Apabila mode yang dipilih tidak tepat maka proses pengontrolan tidak akan berlangsung sesuai harapan.

Jika gain statis proses -  $K$  bernilai positif (proses *direct*) maka mode aksi kontroler *closed loop* yang harus digunakan adalah mode *reverse*, sedangkan jika gain statis proses -  $K$  bernilai negative (proses *reverse*), mode aksi kontroler yang digunakan harus mode *direct*. Secara fungsional prinsip kerja kedua mode aksi tersebut dapat diilustrasikan seperti pada gambar 1.14 berikut:



Gambar 1.14. Prinsip kerja mode aksi reverse (a) dan mode aksi direct (b)

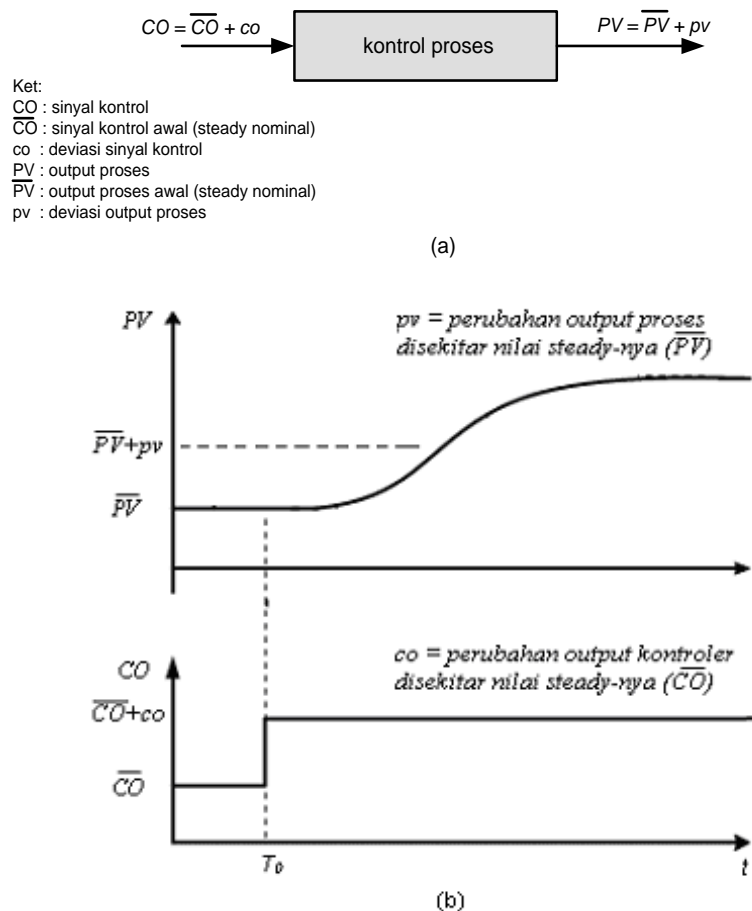
Sebagai ringkasan, berikut adalah prosedur yang harus dilakukan dalam melakukan eksperimen *bump test* atau uji sinyal *step* pada proses FOPDT yang akan/sedang berlangsung:

1. Pastikan kontroler (DCS atau PLC) berada pada mode kontrol manual (*open loop*) dan variabel output proses- $PV$  telah berada disekitar level operasi atau titik kerja yang diharapkan yaitu dengan terlebih dulu men-*setting* nilai  $CO$  secara manual sedemikian sehingga output  $PV$  berada disekitar nilai terharap (*setpoint- $SP$* )

2. Sebelum dilakukan *bump test*, pastikan proses telah berada dalam keadaan *steady* (output proses sudah tidak mengalami dinamika atau sudah menetap disekitar nilai terharap tersebut)
3. Selama perekaman data output proses (eksperimen berlangsung) pastikan tidak terjadi perubahan beban atau terjadi gangguan pada proses. Jika pada saat perekaman data terjadi gangguan beban yang relative besar maka eksperimen perlu diulang lagi
4. Ubah output kontroler dari nilai *steady* sebelumnya menjadi nilai yang baru secara mendadak (*step*). Perubahan *step* pada *CO* ini dapat bernilai positif (nilai  $CO_1 > CO_0$ ) atau negative (nilai  $CO_1 < CO_0$ ). Agar perubahan nilai *CO* ini tidak mengganggu aspek *safety* serta tidak menyebabkan penurunan kualitas produk operasi proses secara ekstrim (*upset*), maka pastikan perubahan *CO* tidak terlalu besar (umumnya sekitar 5%-10%).

#### 1.4.1. Representasi proses FOPDT dengan Fungsi Alih

Representasi atau model matematis sebuah proses penting diketahui selain untuk kepentingan analisis yang lebih mendalam, terutama juga untuk kepentingan simulasi dengan komputer (dalam hal ini misal dengan Simulink).



**Gambar 1.15.** (a) Diagram blok kontrol proses FOPDT (b) Relasi input/output proses tersebut disekitar nilai *steady* nominalnya

**Catatan Penting tentang pengetahuan model Matematis proses bagi operator lapangan**

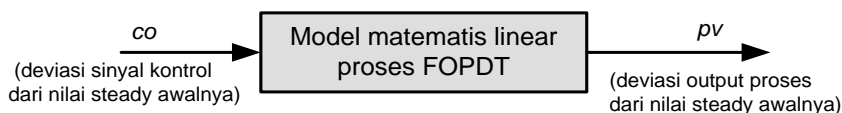
Terkait dengan pengontrolan variabel proses dengan menggunakan kontrol PID, Seorang operator lapangan secara praktis tidak harus mengetahui representasi atau persamaan matematis dari proses yang akan dikontrol. Jika proses yang dihadapi memiliki karakteristik model linear FOPDT, maka operator cukup melakukan eksperimen uji sinyal tangga seperti yang telah dijelaskan sebelumnya ( yaitu untuk mendapatkan parameter-parameter proses  $L$ ,  $T$  dan  $K$ ). Berdasarkan ketiga parameter proses tersebut, konstanta-konstanta kontroler dapat dicari dengan menggunakan metode-metode tuning PID yang telah tersedia (lihat **bab 2**).

Sebelum membahas lebih jauh tentang representasi matematis dari sebuah kontrol proses FOPDT ini, terlebih dulu tinjau kembali diagram blok serta grafik hubungan input proses (output kontroler- $CO$ ) dengan output proses ( $PV$ ) untuk model tersebut pada gambar 1.15.

Berdasarkan gambar 1.15, sinyal output kontroler ( $CO$ ) pada dasarnya dapat diuraikan menjadi dua bagian: sinyal output kontroler *steady* awal ( $\overline{CO}$ ) serta deviasi atau perubahan disekitar nilai *steady*-nya ( $co$ ). Hal yang sama berlaku juga pada variabel output proses ( $PV$ ).

Seperti yang telah dibahas sebelumnya, parameter proses (seperti delay transportasi, gain statis dan konstanta waktu proses) secara praktis dapat dicari secara mudah berdasarkan grafik pada gambar 1.15 (b). Tapi satu hal yang perlu ditekankan, nilai ketiga parameter proses hasil identifikasi pada kenyataannya hanya valid untuk kondisi beban serta kisaran nilai sinyal output kontroler- $CO$  saat eksperimen dilakukan saja. Untuk kondisi beban dan/atau kisaran nilai sinyal kontrol yang berbeda, dinamika prosesnya-pun umumnya akan berbeda juga. Hal ini muncul dari kenyataan bahwa dalam daerah kerja yang luas (misal jangkauan output kontrol  $CO$  antara 0-100%), model kontrol proses itu sendiri bersifat nonlinear.

Dengan alasan tersebut, model matematis linear yang menggambarkan dinamika sebuah proses pada dasarnya hanya terbatas pada kisaran kondisi yang tidak jauh berbeda dengan saat percobaan dilakukan saja, baik besar beban maupun besar sinyal kontrolnya (ingat model FOPDT adalah sebuah model matematis linear). Sehingga agar modelnya realistik, variabel input dan output yang terlibat dalam sebuah proses secara matematis harus direpresentasikan dalam bentuk variabel deviasi (perubahan kecil) disekitar titik kerja *steady*-nya (lihat gambar 1.16 dan catatan penting tentang linearisasi proses/sistem nonlinear).



**Gambar 1.16.** Relasi antara perubahan output proses terhadap perubahan nilai input disekitar nilai *steady*-nya

Untuk model proses FOPDT itu sendiri, dinamika perubahan atau deviasi output proses terhadap perubahan sinyal kontrol disekitar nilai *steady* nominalnya dapat direpresentasikan dalam bentuk persamaan diferensial linear berikut:

$$T \frac{dpv(t)}{dt} = -pv(t) + K.co(t - L), \quad pv(0) = 0 \quad (1.3)$$

dengan :

$K$  : gain statis proses

$L$  : delay transportasi proses

$T$  : konstanta waktu proses

$co$  : deviasi (perubahan) input proses atau output kontroler disekitar nilai *steady*-nya (lihat gambar 1.15 dan 1.16)

$pv$  : deviasi (perubahan) output proses disekitar nilai *steady*-nya (lihat gambar 1.15 dan 1.16)

$dpv/dt$  : laju perubahan output proses disekitar nilai *steady*-nya

$pv(0)$  : deviasi output proses awal sebelum terjadi perubahan input. Karena proses sebelum terjadi perubahan input telah dalam keadaan *steady* (menetap), maka tentu saja nilai deviasi awalnya akan sama dengan nol.

Untuk tujuan analisis dan perancangan (terutama juga untuk kemudahan melakukan uji simulasi dengan Simulink), persamaan diferensial yang menggambarkan perilaku proses diatas secara umum dapat direpresentasikan kedalam bentuk fungsi alih proses seperti yang direlasikan oleh persamaan 1.4 berikut:

$$H(s) = \frac{pv(s)}{co(s)} = \frac{K}{Ts + 1} e^{-sL} \quad (1.4)$$

dengan

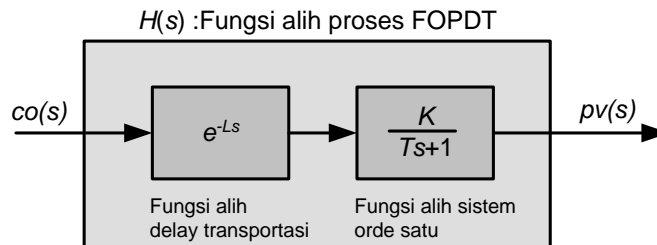
$H(s)$  : fungsi alih kontrol proses FOPDT

$pv(s)$  : transformasi laplace deviasi output proses

$co(s)$  : transformasi laplace deviasi output kontroler (input proses)

$s$  : variabel laplace

Dalam bentuk diagram blok, fungsi alih kontrol proses FOPDT tersebut dapat dipandang sebagai susunan serial dua buah blok fungsional: Elemen delay transportasi dan sistem/proses orde satu seperti terlihat pada gambar 1.17 berikut:



**Gambar 1.17.** Diagram blok model proses FOPDT

#### Catatan penting tentang linearisasi proses/sistem nonlinear

Sebuah proses non linear pada dasarnya dapat direpresentasikan dalam bentuk persamaan diferensial berikut:

$$\frac{dPV}{dt} = f(PV, CO)$$

dalam hal ini  $CO$  dan  $PV$  berturut-turut adalah input dan output proses, sedangkan  $f(.)$  adalah sebuah fungsi non linear.

Dengan menganggap proses telah menetap pada nilai *steady* tertentu yaitu  $\overline{CO}$  dan  $\overline{PV}$ , maka jika satu saat input  $CO$  mengalami deviasi kecil disekitar nilai *steady*-nya:

$$CO(t) = \overline{CO} + co(t)$$

perubahan input tersebut secara langsung akan menyebabkan perubahan atau deviasi pada output proses ( $PV$ ):

$$PV(t) = \overline{PV} + pv(t)$$

Dengan menggunakan deret Taylor, dapat dibuktikan bahwa persamaan proses disekitar nilai *steady*-nya tersebut dapat didekati oleh persamaan diferensial linear berikut

$$\frac{dpv(t)}{dt} = A.pv(t) + B.co(t)$$

dalam hal ini  $pv(t)$  dan  $co(t)$  berturut-turut adalah variable deviasi (perubahan) disekitar nilai *steady* input dan output proses tersebut. Sedangkan A dan B merupakan konstanta-konstanta proses yang dapat dicari secara eksperimental (misal dengan eksperimen *bump test* yang telah dijelaskan sebelumnya).

Perhatikan bahwa jika dalam keadaan *steady* awal, sinyal output kontrol dan output proses keduanya bernilai nol, maka tentu saja nilai deviasi kedua variable tersebut secara langsung akan menunjukkan nilai nominalnya:

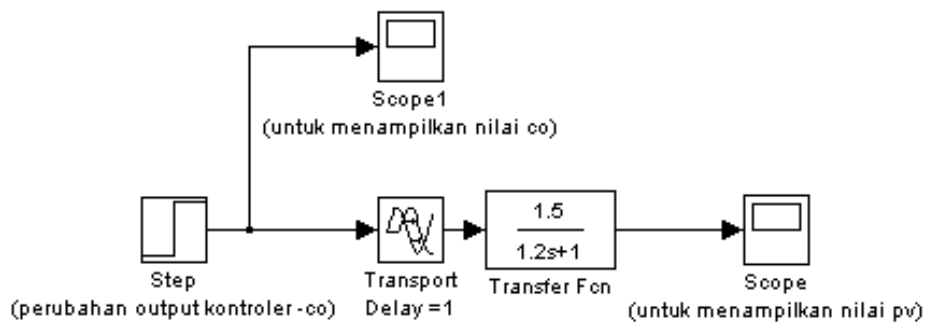
$$CO(t) = co(t)$$

$$PV(t) = pv(t)$$

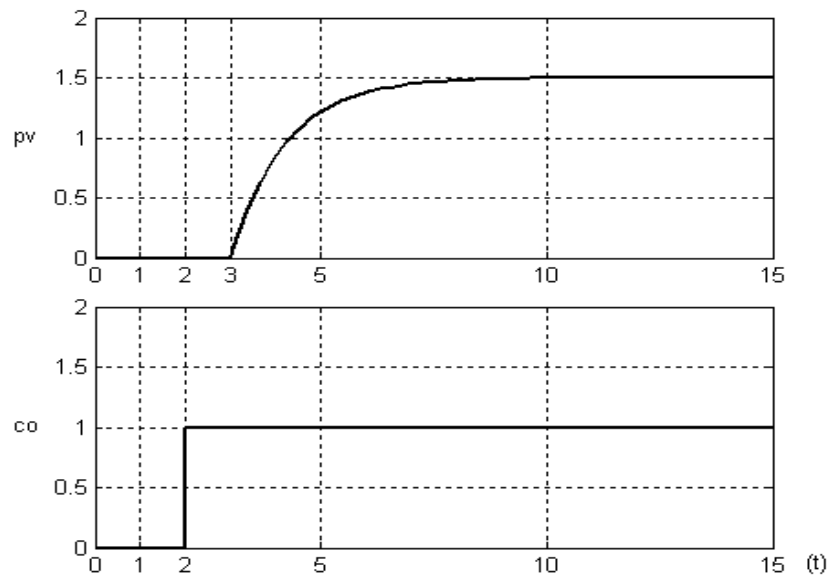
Di dalam Simulink, dinamika kontrol proses FOPDT pada dasarnya dapat disimulasikan secara mudah dengan menggunakan blok-blok fungsional (dalam bentuk fungsi alih) yang telah disediakan pada perangkat lunak tersebut. Sebagai contoh, gambar 1.18 dan 1.19 berikut berturut-turut memperlihatkan blok simulink dan relasi input-outputnya untuk dinamika sebuah proses FOPDT dengan persamaan:

$$pv(s) = \frac{1.5}{1.2s + 1} e^{-s} . co(s) \quad ; (K=1.5, T=1.2, L=1)$$





**Gambar 1.18.** Contoh blok simulasi dengan Simulink



**Gambar 1.19.** Hasil simulasi untuk gambar 1.18

#### 1.4.2. Akomodasi Model Gangguan pada Kontrol Proses FOPDT

Pada **subbab 1.4.1** telah dibahas dinamika atau fungsi alih antara deviasi output proses terhadap perubahan sinyal input kontrol disekitar nilai *steady*-nya. Untuk studi analisis yang lengkap, tentunya perlu diketahui juga pengaruh perubahan variabel *exogenous* (input gangguan) terhadap dinamika output proses.

Seperti halnya hubungan antara perubahan output kontroler (*co*) terhadap dinamika perubahan output proses (*pv*), hubungan antara input gangguan (perubahan variabel *exogenous*) terhadap dinamika perubahan output proses (*pv*) untuk model FOPDT dalam banyak kasus umumnya akan menunjukkan karakteristik dinamika FOPDT juga (lihat persamaan 1.5).

$$H_d(s) = \frac{pv(s)}{d(s)} = \frac{K_d}{T_d s + 1} e^{-sL_d} \quad (1.5)$$

dalam hal ini:

$H_d(s)$  : fungsi alih gangguan untuk proses FOPDT

$K_d$  : Gain statis gangguan

$T_d$  : Konstanta waktu gangguan

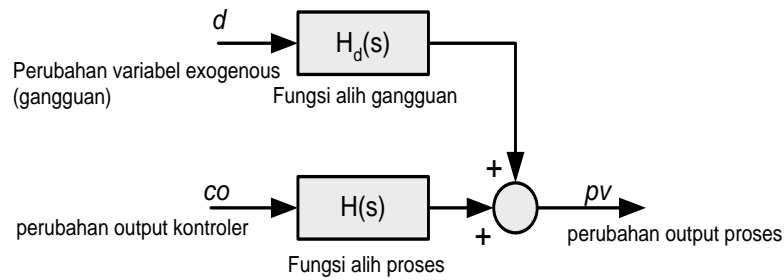
$L_d$  : Keterlambatan transportasi gangguan

Jika pengaruh input gangguan ( $d$ ) diatas diperhitungkan bersama-sama dengan perubahan sinyal kontrol ( $co$ ), maka deviasi output proses pada dasarnya merupakan penjumlahan linear dari kedua jenis input tersebut, atau secara matematis dapat ditulis :

$$pv(s) = H(s).co(s) + H_d(s).d(s) \quad 1.6.a$$

$$pv(s) = \frac{K}{TS + 1} e^{-Ls} .co(s) + \frac{K_d}{T_d s + 1} e^{-L_d s} .d(s) \quad 1.6.b$$

Dalam bentuk diagram blok, persamaan 1.6 diatas akan nampak seperti terlihat pada gambar 1.20 berikut ini:

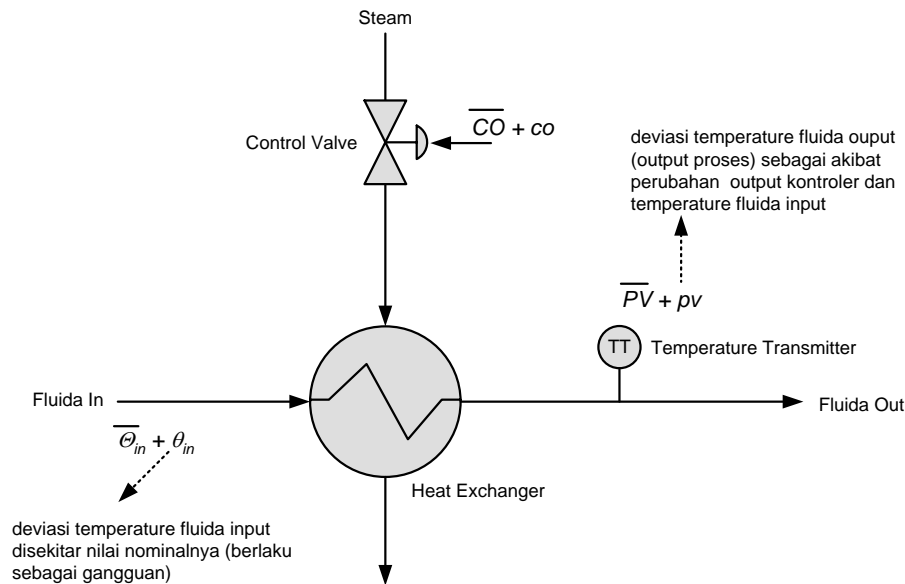


**Gambar 1.20.** Blok diagram relasi output proses dengan sinyal kontrol dan input gangguan

Sebagai studi kasus, tinjau sistem pemanasan fluida pada *heat exchanger* gambar 1.21. Dalam sistem kontrol tersebut, kita anggap satu-satunya gangguan yang mempengaruhi variabel output proses (dalam hal ini temperature fluida output) adalah terjadinya variasi temperature fluida input disekitar nilai nominalnya (perhatikan untuk sistem yang ditinjau, gangguan juga dapat disebabkan oleh deviasi atau perubahan debit fluida input, perubahan tekanan *steam*, perubahan temperature lingkungan, dan sebagainya. Tapi dalam kasus ini, semua variabel tersebut dianggap konstan ).

Seperti yang telah dibahas pada **subbab 1.4.1.**, hubungan antara perubahan output controller -  $co$  terhadap perubahan temperature fluida output -  $pv$  untuk kasus diatas pada dasarnya akan menunjukan dinamika model FOPDT.

Sedangkan hubungan antara perubahan temperature fluida input -  $\theta_{in}$  (gangguan) terhadap perubahan fluida output -  $pv$  analisisnya adalah sebagai berikut: Terlebih dulu anggap tidak ada perubahan sinyal kontrol ( $co = 0$ ) untuk proses tersebut, jika misalnya satu saat temperature fluida input berubah tiba-tiba dari  $\bar{\Theta}_{in}$  menjadi  $\bar{\Theta}_{in} + \theta_{in}$ , maka temperature fluida output juga perlahan-lahan akan mengalami deviasi dari nilai *steady* sebelumnya -  $\bar{PV}$  menjadi  $\bar{PV} + pv$  (lihat gambar 1.21).



**Gambar 1.21.** Gangguan berupa perubahan temperature fluida input disekitar nilai nominalnya

Untuk kasus ini, dapat dibuktikan secara matematis (lihat **subbab 1.6 Contoh Soal dan penyelesaian**), fungsi alih antara perubahan temperature fluida output ( $pv$ ) sebagai akibat perubahan temperature fluida input  $\theta_{in}$  (dalam hal ini sebagai input gangguan) adalah:

$$H_d(s) = \frac{pv(s)}{d(s)} = \frac{pv(s)}{\theta_{in}(s)} = \frac{1}{T_d s + 1} e^{-sL_d} \quad (1.7)$$

dengan

$H_d(s)$  : fungsi alih gangguan

$pv$  : deviasi variabel proses (perubahan temperature fluida output)

$d = \theta_{in}$  : deviasi temperature fluida input (input gangguan)

$T_d$  : konstanta waktu gangguan (nilainya sama dengan  $T$  pada persamaan 1.3)

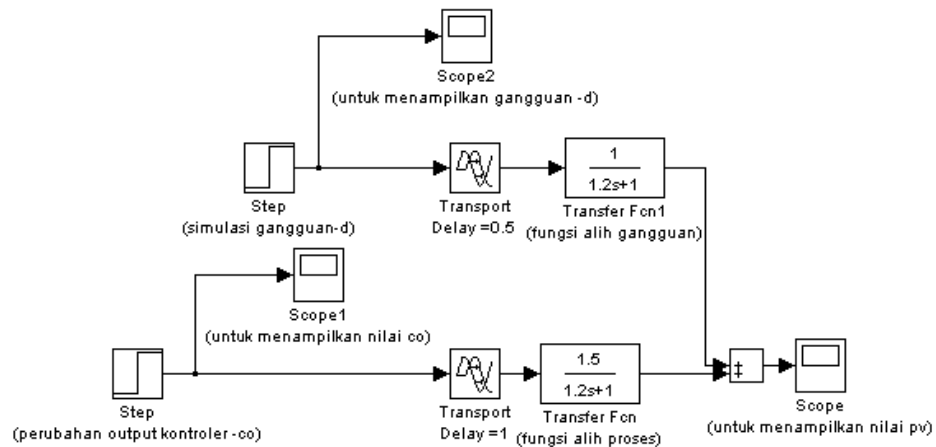
$L_d$  : delay transportasi gangguan (nilainya tergantung pada panjang pipa dan laju aliran fluida)

Seperti nampak pada persamaan 1.7, dinamika perubahan temperature fluida output (variabel output) sebagai akibat variasi atau perubahan temperature fluida input (gangguan) pada dasarnya akan menunjukkan dinamika model gangguan FOPDT yang ekuivalen dengan persamaan 1.5 (dalam hal ini gain statis gangguan  $K_d$  untuk kasus diatas nilainya selalu sama dengan satu, artinya dalam keadaan *steady* yang baru, deviasi temperature fluida output yang disebabkan oleh perubahan temperature fluida input, besarnya akan selalu sama dengan deviasi temperature fluida input tersebut).

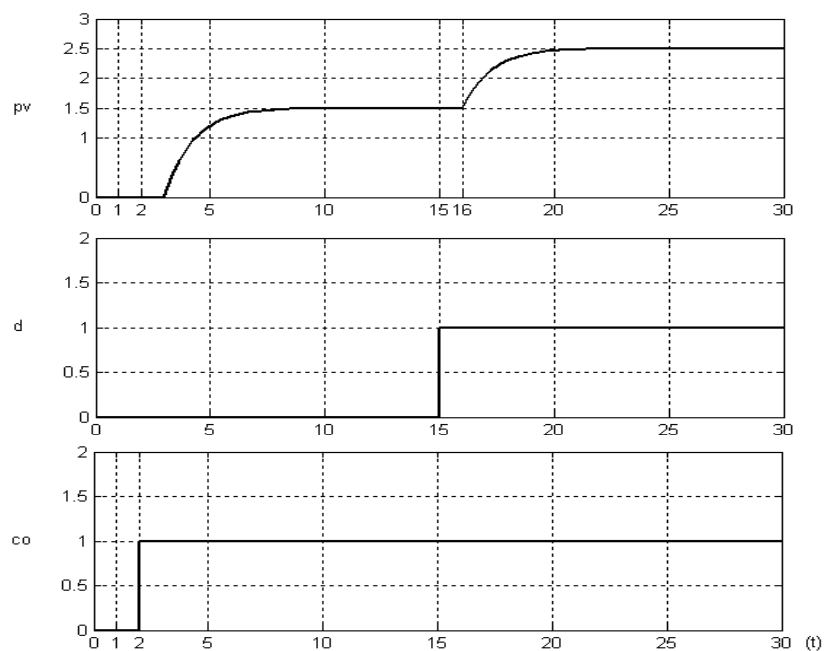
Untuk lebih memahami pengaruh perubahan output kontroler  $-co$  dan pengaruh input gangguan  $-d$  terhadap dinamika output sebuah model proses FOPDT, maka akan sangat membantu jika kita secara langsung mengujinya dalam bentuk simulasi komputer.

Gambar 1.22 dan 1.23 berikut berturut-turut memperlihatkan blok Simulink dan relasi input-outputnya untuk sebuah proses yang memiliki dinamika dengan persamaan :

$$pv(s) = \frac{1.5}{1.2s+1} e^{-s} .co(s) + \frac{1}{1.2s+1} e^{-0.5s} .d \quad ;(K=1.5, \quad T=T_d=1.2, \quad K_d=1, \quad L=1, \quad L_d=0.5)$$



**Gambar 1.22.** Contoh blok simulasi Simulink untuk model proses dengan gangguan

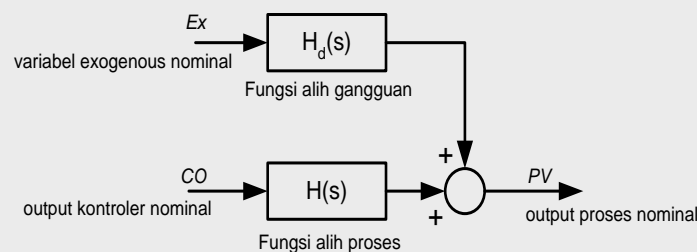


**Gambar 1.23.** Hasil simulasi untuk gambar 1.22

**Catatan penting tentang model proses Linear untuk kemudahan analisi dan simulasi**

Jika semua variabel input penting yang mempengaruhi output sebuah proses telah diakomodasi dalam persamaan model matematisnya, dan proses tersebut kita anggap linear untuk seluruh daerah kerjanya (hal ini dimaksudkan untuk memudahkan analisis), maka variabel input-output pada model proses pada dasarnya dapat ditulis langsung sebagai variabel nominalnya (bukan variabel deviasi) –lihat persamaan 1.8 dan blok diagramnya pada gambar 1.24.

$$PV(s) = H(s).CO + H_d(s).Ex(s) \quad (1.8)$$



**Gambar 1.24.** Diagram blok Proses Linear

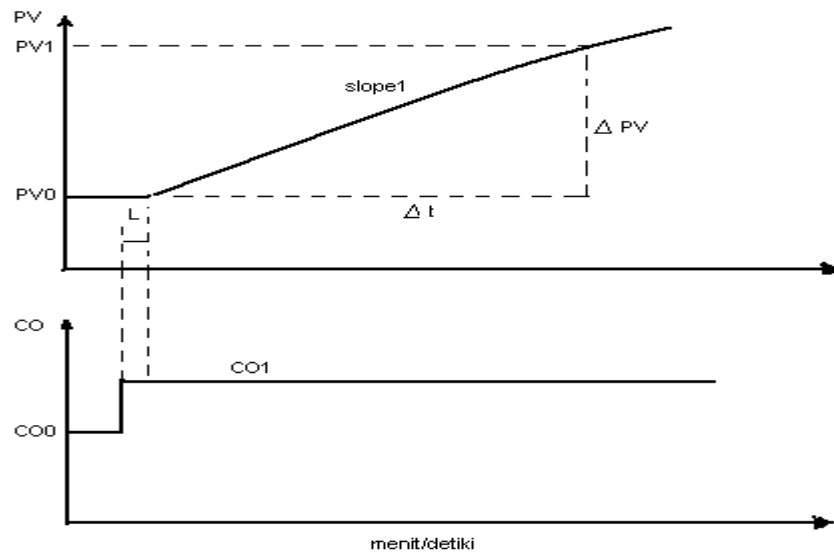
Sebagai contoh, jika gambar 1.24 dimaksudkan untuk menggambarkan dinamika sistem *heat exchanger* pada gambar 1.21, maka variabel input-output proses: *CO*, *Ex* dan *PV* pada gambar 1.24 (atau persamaan 1.8) diatas berturut-turut pada dasarnya dapat dianggap langsung sebagai output kontroler nominal, temperature fluida input, dan temperature fluida output yang terukur sensor (**bukan deviasinya**).

Perlu ditekankan bahwa persamaan 1.8 atau blok diagram pada gambar 1.24 diatas dapat juga digeneralisasi untuk berbagai model proses lainnya (seperti model IPDT-lihat **subbab 1.5**).

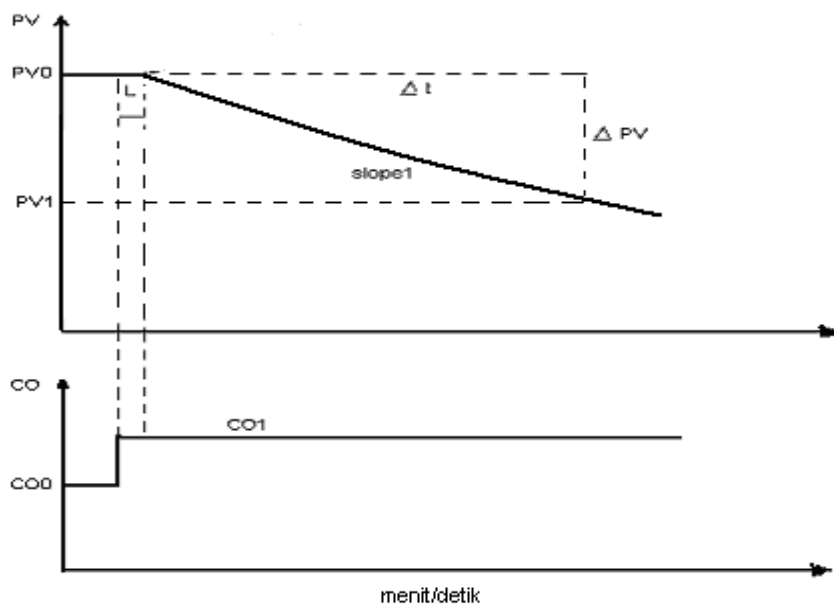
### 1.5. Model Non Self Regulating Process (Model IPDT)

Sebuah proses dapat dikategorikan sebagai model *Non Self Regulating* jika hubungan antara input-output proses tersebut bersifat tidak stabil. Seperti yang telah sedikit disinggung pada **subbab Pendahuluan**, salah satu model *Non Self Regulating* yang umum dijumpai di industri adalah model IPDT (*Integrating Plus Dead Time*).

Proses dapat didekati oleh model matematis IPDT jika untuk sebuah perubahan tangga input kontrol - *CO* pada keadaan *steady* atau titik kesetimbangannya, output proses – *PV* tersebut akan terus menerus membesar atau terus menerus mengecil tergantung dari sifat prosesnya itu sendiri (apakah bersifat *direct* atau *reverse*). Gambar 1.25 dan 1.26 berikut berturut-turut memperlihatkan tanggapan tipikal output model proses IPDT *direct* dan model proses IPDT *reverse* terhadap perubahan input kontrol disekitar nilai kesetimbangannya.



**Gambar 1.25.** Relasi input-output untuk model IPDT (jenis proses: *direct*)



**Gambar 1.26.** Relasi input-output untuk model IPDT (jenis proses: *reverse*)

Untuk kepentingan perancangan dan tuning kontroler-nya, hubungan input-output seperti yang nampak pada gambar 1.25 (atau gambar 1.26) pada dasarnya dapat dicirikan hanya oleh dua parameter proses saja: *process transport delay* ( $L$ ) dan *process integrative Gain* ( $K^*$ ).

- **Process transport delay (keterlambatan transportasi proses) -  $L$ :** Keterlambatan output proses dalam menanggapi perubahan output kontroler (penjelasan teknisnya hampir sama dengan proses FOPDT)
- **Process integrative gain (Gain integratif proses) -  $K^*$ :** Perbandingan antara perubahan kemiringan (*slope*) output proses dengan besar perubahan (*magnitude*) sinyal kontroler. Jika keadaan awal proses telah berada dalam titik kesetimbangan atau keadaan *steady*-nya (keadaan kesetimbangan ini secara praktis pada

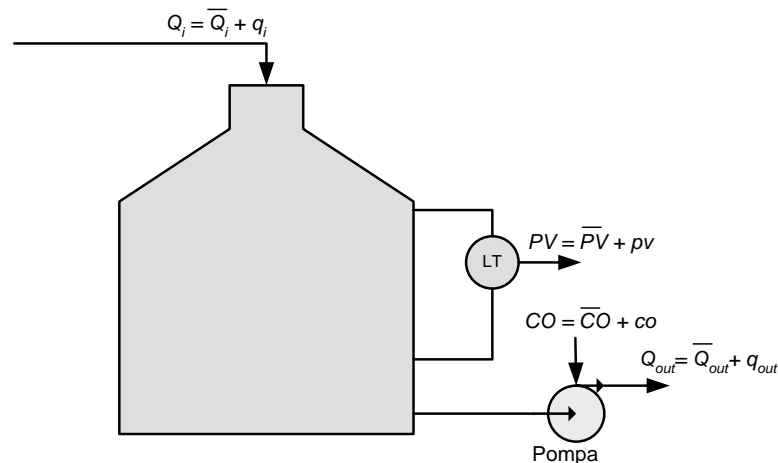
kenyataannya sangat sulit dicapai) maka berdasarkan grafik pada gambar 1.25 atau 1.26, besarnya gain integrative dapat dirumuskan :

$$K^* = \frac{\Delta slope}{\Delta CO} = \frac{slope_1}{\Delta CO} = \frac{(PV_1 - PV_0) / \Delta t}{CO_1 - CO_0} \quad 1.9$$

Dalam hal ini  $K^*$  akan bernilai positive jika prosesnya bersifat *direct* seperti pada gambar 1.25 dan akan bernilai negative jika prosesnya bersifat *reverse* seperti pada gambar 1.26 (Ingat pengetahuan sifat proses ini sangat penting untuk menentukan mode kontroler PID yang harus dipilih: *Reverse* atau *Direct* – lihat kembali **subbab. 1.4**).

Seperti halnya model proses FOPDT, besarnya gain integrative pada proses IPDT secara langsung pada dasarnya menunjukkan kesensitifan model proses tersebut : semakin besar nilai gain integratif, maka proses IPDT akan semakin sensitif.

Salah satu contoh proses nyata di industri untuk model IPDT ini adalah pengontrolan level fluida pada tangki dengan menggunakan pompa penghisap sebagai penggerak pada sisi outletnya (lihat gambar 1.27)



**Gambar 1.27.** Pengontrolan level tanki lewat pompa

Dalam kasus pengontrolan level tersebut, debit fluida output pada dasarnya hanya tergantung pada besar sinyal kontrol penggerak pompa (tidak tergantung terhadap level fluida), hal ini akan berbeda jika penggerak yang digunakan adalah *control valve*- lihat **Subbab 1.6 soal dan penyelesaian**.

Dengan menganggap tidak terjadi perubahan debit fluida input ( $q_{in} = 0$ ) serta sistem telah berada dalam kesetimbangannya (debit fluida input -  $Q_{in}$  = debit fluida output -  $Q_{out}$ ), maka perubahan level fluida di dalam tangki (dalam hal ini berlaku sebagai output proses -  $p_v$ ) tentunya hanya tergantung pada perubahan debit fluida output yang disebabkan oleh perubahan sinyal kontrol penggerak pompa ( $co$ ): Jika sinyal kontrol diperkecil dari keadaan *steady* awalnya, maka debit fluida output juga akan semakin mengecil, semakin kecil debit fluida output maka level fluida akan semakin naik (hal sebaliknya terjadi jika sinyal kontrol diperbesar).

Disekitar nilai *steady* atau titik kesetimbangannya tersebut, dinamika perubahan input-output proses diatas secara matematis dapat direpresentasikan dalam bentuk persamaan diferensial linear berikut:

$$\frac{dpv(t)}{dt} = K^* \cdot co(t - L) \quad , pv(0)=0 \quad 1.10$$

dalam hal ini:

$K^*$  : gain integratif proses

$L$  : delay transportasi proses

$co$  : deviasi (perubahan) input proses atau output kontroler disekitar nilai *steady*-nya

$p_v$  : deviasi (perubahan) output proses disekitar nilai *steady*-nya

$dpv/dt$ : laju perubahan output proses disekitar nilai *steady*-nya

$p_v(0)$  : deviasi output proses awal sebelum terjadi perubahan input. Karena proses sebelum terjadi perubahan input telah dalam keadaan *steady* (menetap), maka tentu saja nilai deviasi awalnya akan sama dengan nol.

Jika disajikan dalam bentuk fungsi alihnya, maka persamaan 1.10 akan berbentuk:

$$H(s) = \frac{pv(s)}{co(s)} = \frac{K^*}{s} e^{-sL} \quad 1.11$$

Perlu diperhatikan disini bahwa untuk model proses kontrol level dengan pompa ini, gain integratifnya ( $K^*$ ) akan bernilai negative (level fluida akan semakin menurun untuk setiap perubahan positive pada output kontrol penggerak pompa)

Sekarang, bagaimana jika debit fluida input untuk proses diatas mengalami perubahan dari nilai *steady* awalnya ?, Dalam kasus ini perubahan debit fluida input pada dasarnya berlaku sebagai input gangguan-*disturbance*. Dengan cara analisis yang relative sama seperti penjelasan sebelumnya, dapat dibuktikan bahwa secara matematis hubungan antara perubahan debit fluida input terhadap perubahan level fluida untuk kasus tersebut pada dasarnya juga akan memiliki model IPDT :

$$\frac{pv(s)}{d(s)} = \frac{pv(s)}{q_{in}(s)} = \frac{K_d^*}{s} e^{-sL_d} \quad 1.12$$

Dalam hal ini nilai gain integratif gangguan ( $K_d^*$ ) pada persamaan 1.12 akan bernilai positive (level fluida akan semakin naik untuk setiap perubahan positive pada debit fluida input).

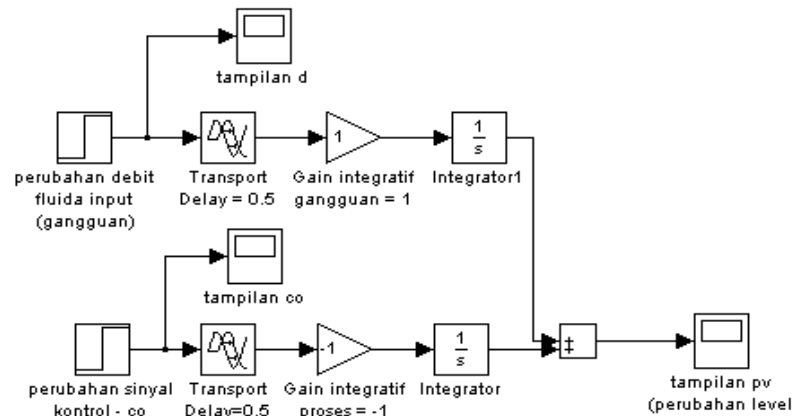
Dengan menggabungkan persamaan 1.11 dan 1.12, maka dinamika perubahan level fluida karena pengaruh perubahan output kontroler dan perubahan debit fluida input (sebagai gangguan) secara lengkap dapat direpresentasikan seperti terlihat pada persamaan 1.13 (Persamaan tersebut pada dasarnya juga berlaku untuk model IPDT lainnya).

$$\frac{pv(s)}{d(s)} = \frac{pv(s)}{q_{in}(s)} = \frac{K^*}{s} e^{-sL} co(s) + \frac{K_d^*}{s} e^{-sL_d} d(s) \quad 1.13$$

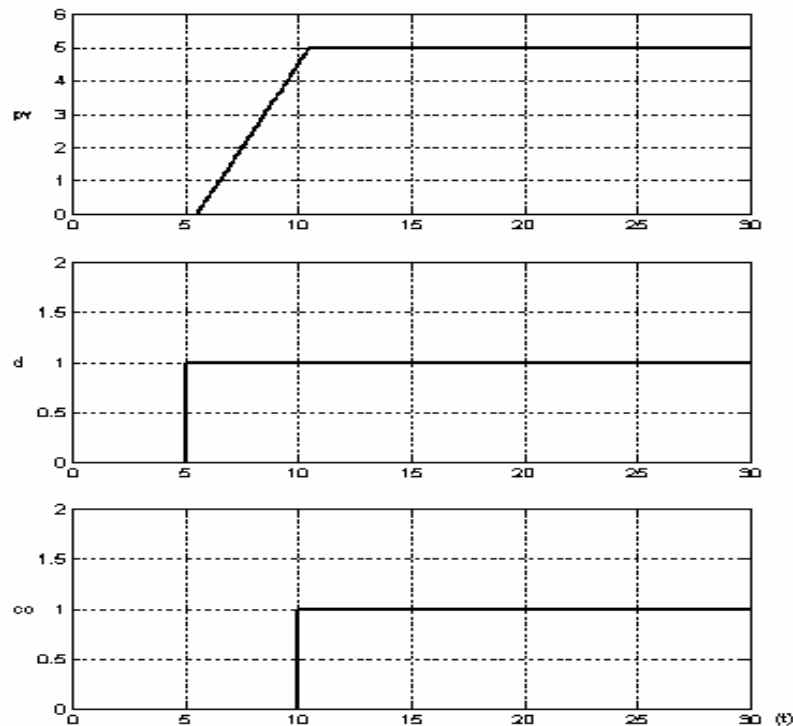


Gambar 1.28 dan 1.29 dibawah ini berturut-turut memperlihatkan contoh blok Simulink dan salah satu hasil simulasi untuk model proses IPDT (misal dalam hal ini sistem kontrol level pada gambar 1.27) dengan persamaan dinamika berikut:

$$pv(s) = \frac{-1}{s} e^{-s0.5} .co(s) + \frac{1}{s} e^{-s0.5} q_{in}(s) \quad ; (K^* = -1, K_d^* = 1, L = L_d = 0.5)$$



**Gambar 1.28.** Blok diagram Simulink untuk model IPDT



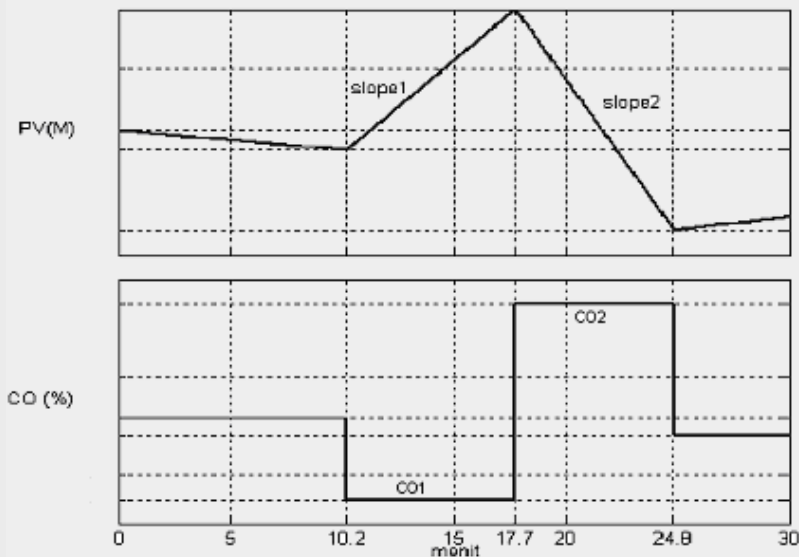
**Gambar 1.29.** Hasil simulasi untuk gambar 1.28

### Catatan penting tentang Identifikasi model IPDT secara praktis

Terkait dengan sifat prosesnya yang tidak stabil, Khusus untuk model proses IPDT (atau proses integrative lainnya), identifikasi parameter proses pada dasarnya sulit dilakukan dalam keadaan *steady*-nya. Hal ini disebabkan secara praktis kita sangat sukar membawa proses IPDT pada keadaan setimbangnya. Misal dalam kasus pengontrolan level tangki – gambar 1.27, operator umumnya tidak tahu nilai pasti sinyal kontrol nominal sedemikian sehingga titik kesetimbangan tercapai (debit fluida output sama dengan debit input).

Oleh sebab itu maka dalam prakteknya, untuk memperoleh parameter proses IPDT ( $K^*$  dan  $L$ ), percobaan bump test tidak perlu dilakukan pada keadaan yang betul-betul *steady*, eksperimen tersebut cukup dilakukan disekitar titik kerja proses tersebut. Tetapi untuk mendapatkan nilai gain integratif, perubahan sinyal kontrol secara praktis perlu dilakukan minimal dua kali.

Gambar 1.30 berikut ini memperlihatkan ilustrasi hubungan input-output model proses IPDT jenis *reverse* (dalam hal ini misal kontrol level fluida dengan pompa) hasil dari eksperimen Bump Test.



**Gambar 1.30.** Relasi input-output untuk model IPDT (jenis proses: reverse)

Berdasarkan definisinya, maka dari gambar 1.30 tersebut, nilai gain integratif dapat dihitung berdasarkan persamaan 1.14 berikut:

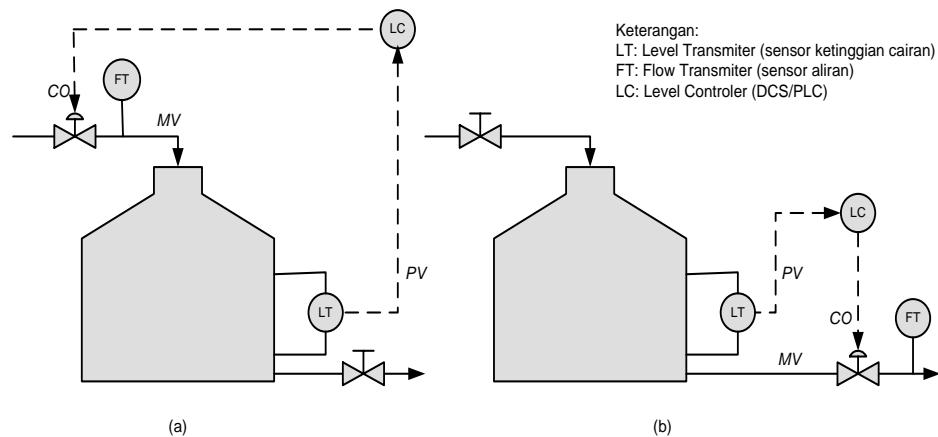
$$K^* = \frac{\Delta \text{slope}}{\Delta CO} = \frac{\text{slope}_2 - \text{slope}_1}{CO_2 - CO_1} \quad 1.14$$

## 1.6. Contoh Soal dan Penyelesaian

### Contoh 1.1

#### Pemilihan mode aksi *loop* tertutup pada kontroler komersil (1)

Gambar 1.31 dibawah ini memperlihatkan dua buah alternative pengontrolan level fluida pada sebuah tanki dengan menggunakan *control valve*



**Gambar 1.31.** Dua alternative pengontrolan level cairan pada tanki penampung

#### Pertanyaan

Jika *control valve* yang digunakan berjenis ATO (*air to Open*), pilihlah mode aksi kontroler *loop* tertutup pada DCS/PLC yang tepat untuk kedua gambar tersebut

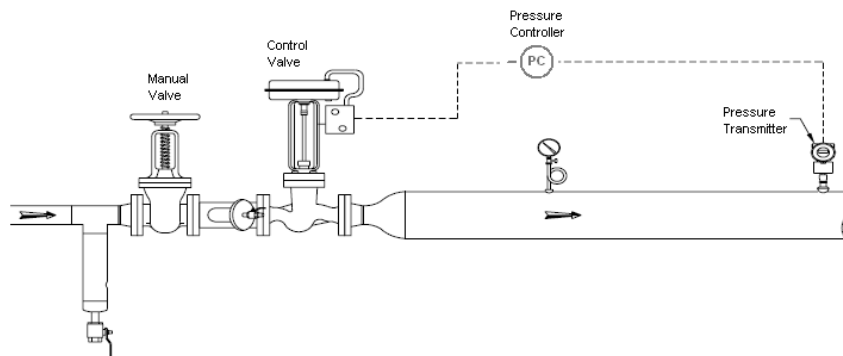
#### Penyelesaian

Untuk proses pada gambar 1.31 (a) mode aksi kontroler yang harus digunakan adalah mode *reverse*, hal ini disebabkan prosesnya bersifat *direct* (perhatikan, semakin besar sinyal *CO*, maka bukaan valve input semakin lebar sehingga level cairan pada tanki semakin tinggi--*PV* semakin besar), sedangkan untuk kasus gambar 1.31 (b) mode aksi kontroler yang harus digunakan adalah mode *direct*, hal ini disebabkan proses yang dikontrol memiliki sifat *reverse* (Semakin besar sinyal *CO*, maka bukaan valve output semakin lebar sehingga level cairan pada tanki semakin turun-- *PV* semakin kecil).

### Contoh 1.2

#### Pemilihan mode aksi *loop* tertutup pada kontroler komersil (2)

Tinjau sistem kontrol tekanan uap panas yang mengalir dalam pipa seperti nampak pada gambar 1.32 berikut:



**Gambar 1.32.** Sistem kontrol tekanan uap pada pipa

### Pertanyaan

Pilihlah mode aksi kontroler *loop* tertutup pada DCS/PLC yang tepat untuk sistem tersebut

### Penyelesaian

Seperti halnya jawaban **contoh soal 1.1**, untuk menentukan mode kontroler yang tepat, maka terlebih dulu kita harus mengetahui sifat proses tersebut: apakah proses bersifat *direct* atau *reverse*. Hal ini pada dasarnya dapat diketahui dari jenis *control valve* yang digunakan: apakah bertipe *Air To Open* --ATO (*fail close*) atau *Air To Close* --ATC (*fail open*): Jika *control valve* yang digunakan bertipe ATO, maka jelas mode kontroler yang harus dipilih adalah mode *Reverse*, hal ini disebabkan hubungan sinyal kontrol dan output proses (dalam hal ini tekanan uap) dalam kasus ini akan bersifat *direct* (perhatikan semakin besar sinyal kontrol, posisi valve akan semakin terbuka – tekanan semakin tinggi), dan sebaliknya jika *control valve* yang digunakan bertipe ATC, maka mode kontroler yang dipilih adalah mode *Direct*

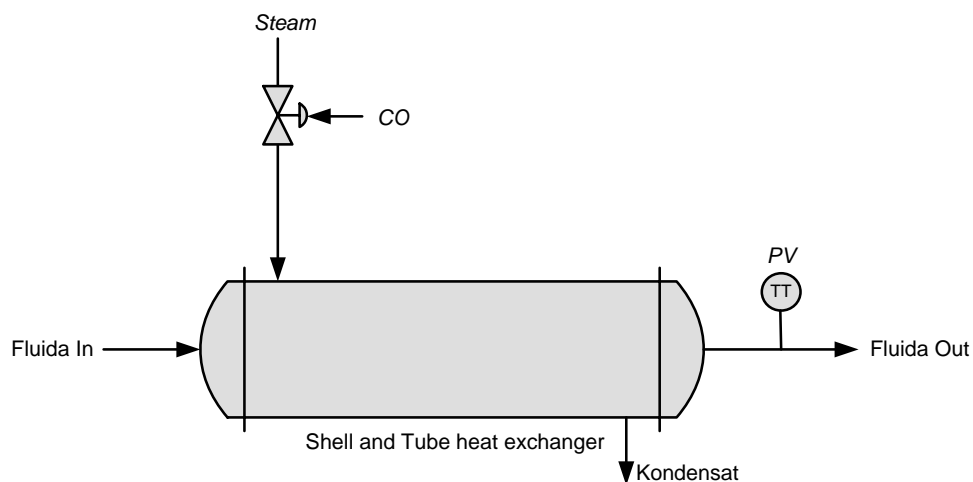
Pengetahuan jenis *control valve* yang digunakan tersebut pada dasarnya dapat dianalisis dari letak valve manual yang disusun seri dengan *control valve* yang akan dimanipulasi oleh kontroler : Karena susunannya serial, maka dalam keadaan normal tentunya manual valve tersebut haruslah terbuka (bersifat normaly open) sehingga aliran uap dapat dikontrol secara otomatis oleh *control valve*. Sedangkan jika terjadi kegagalan operasi pada *control valve*, maka aliran uap tersebut harus dapat dikontrol secara manual oleh operator lewat valve manual tersebut.

Dari hasil analisis diatas, maka dapat dipastikan bahwa ketika terjadi kegagalan operasi maka *control valve* tersebut harus terbuka penuh (atau dengan kata lain jenis *control valve* yang digunakan harus bertipe *fail open* atau ATC), sehingga dengan demikian mode kontroler yang harus dipilih adalah mode *Direct*.

### Contoh 1.3

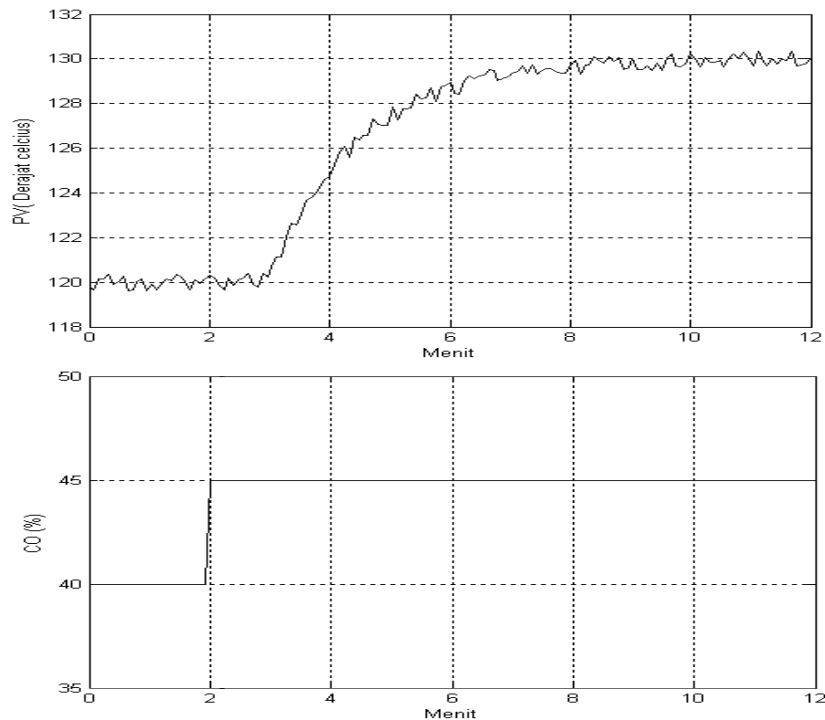
#### Analisis parameter proses FOPDT berdasarkan hasil uji eksperimen bump test (1)

Tinjau sistem *heat exchanger* jenis *Shell and Tube* pada gambar 1.33 berikut



**Gambar 1.33.** *Shell and tube heat exchanger*

Berdasarkan hasil eksperimen *bump test* pada sistem *heat exchanger* diatas, untuk sebuah tekanan *steam* tertentu didapat data hubungan *CO* vs *PV* seperti terlihat pada gambar 1.34 berikut:



**Gambar 1.34.** Hasil eksperimen *bump test* pada sebuah sistem *Heat Exchanger*

#### Pertanyaan

- Berdasarkan grafik relasi dalam gambar 1.34, hitunglah parameter proses ( $L$ ,  $T$  dan  $K$ ) dan sekaligus representasi model matematis hubungan antara perubahan output kontroler terhadap perubahan temperature fluida output pada *heat exchanger* tersebut
- Untuk semua keadaan awal yang sama, berapakah nilai temperature pada keadaan *steady* baru jika operator mengubah sinyal kontrol nominal dari 40% menjadi 60%
- Untuk semua keadaan awal yang sama, berapakah nilai temperature pada keadaan *steady* baru jika operator mengubah sinyal kontrol nominal dari 40% menjadi hanya 35%
- Untuk semua keadaan awal yang sama, berapakah output kontroler yang diperlukan sehingga dalam keadaan *steady* yang baru temperature fluida outlet akan bernilai 180 °C

#### Penyelesaian

- Berdasarkan grafik gambar 1.34, parameter proses tersebut berturut turut (lihat juga gambar 1.35):

-Keterlambatan waktu proses :  $L = 1$  (menit)

Artinya: Proses mulai menanggapi perubahan sinyal  $CO$ , setelah 1 menit berlalu.

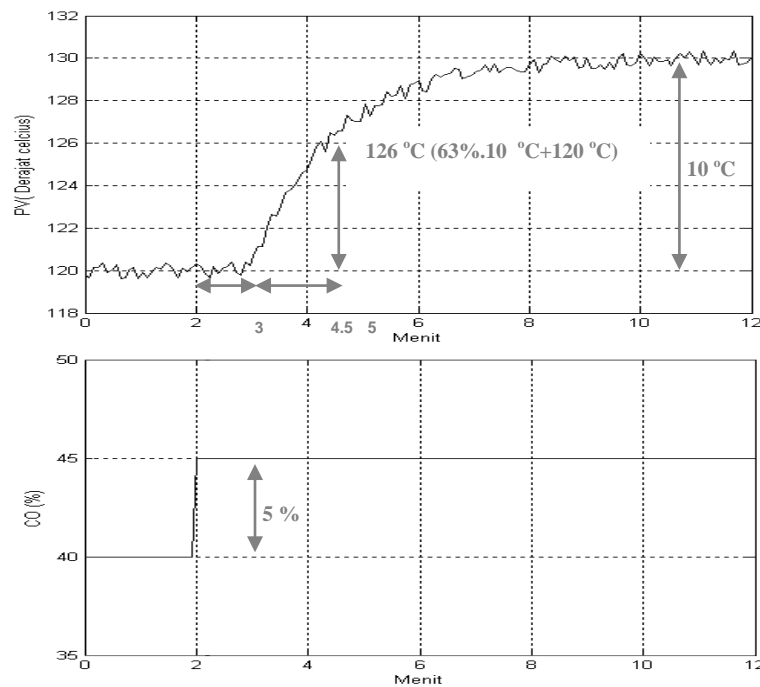
$$\text{-Gain Statis proses : } K = \frac{\Delta PV}{\Delta CO} = \frac{(130-120)^{\circ}C}{(45-40)\%} = 2(^{\circ}C / \%)$$

Artinya: Setiap perubahan sinyal output kontroler ( $CO$ ) sebesar 1%, variabel proses yang dikontrol akan mengalami deviasi sebesar 2 °Celcius.

-Konstanta waktu proses adalah waktu yang diperlukan sehingga perubahan output proses  $PV$  bernilai 63% dari nilai awal *steady*-nya :

$$PV_{63\%} = 63\%.\Delta PV + PV_0 = 126^{\circ}C$$

Berdasarkan grafik, terlihat bahwa nilai 126 °C ini dicapai dalam selang waktu kurang lebih 1.5 menit, sehingga dengan demikian Konstanta waktu proses tersebut adalah 1.5 menit.



**Gambar 1.35.** Perhitungan parameter proses FOPDT berdasarkan hasil uji bump test

Dengan demikian, fungsi alih kontrol proses di daerah kerja tersebut adalah:

$$H(s) = \frac{pv(s)}{co(s)} = \frac{2}{1.5s + 1} e^{-s}$$

**b.** Nilai temperature akhir untuk sebuah perubahan sinyal kontrol tertentu pada dasarnya dapat dicari dengan menggunakan **perumusan 1.2**

$$K = \frac{\Delta PV}{\Delta CO} = \frac{\overline{PV}_1 - \overline{PV}_0}{\overline{CO}_1 - \overline{CO}_0} \text{ atau } \overline{PV}_1 = K(\overline{CO}_1 - \overline{CO}_0) + \overline{PV}_0$$

Berdasarkan hasil jawaban (a) diketahui gain statis proses  $K = 2 (^{\circ}C/\%)$ , sehingga nilai temperature untuk keadaan steady akhir yang baru:

$$\overline{PV}_1 = 2(60 - 40) + 120 = 160^{\circ}\text{Celcius}$$

c. Dengan menggunakan cara yang sama seperti jawaban (b), maka nilai temperature output pada keadaan steady akhir yang baru didapat:

$$\overline{PV}_1 = 2(35 - 40) + 120 = 110 \text{ } ^\circ\text{Celcius}$$

d. Seperti halnya jawaban (b), Besar sinyal kontrol akhir yang diperlukan untuk membawa temperature output pada nilai tertentu pada dasarnya dapat dicari dari relasi 1.2

$$K = \frac{\Delta PV}{\Delta CO} = \frac{\overline{PV}_1 - \overline{PV}_0}{\overline{CO}_1 - \overline{CO}_0} \text{ atau } \overline{CO}_1 = \frac{\overline{PV}_1 - \overline{PV}_0}{K} + \overline{CO}_0$$

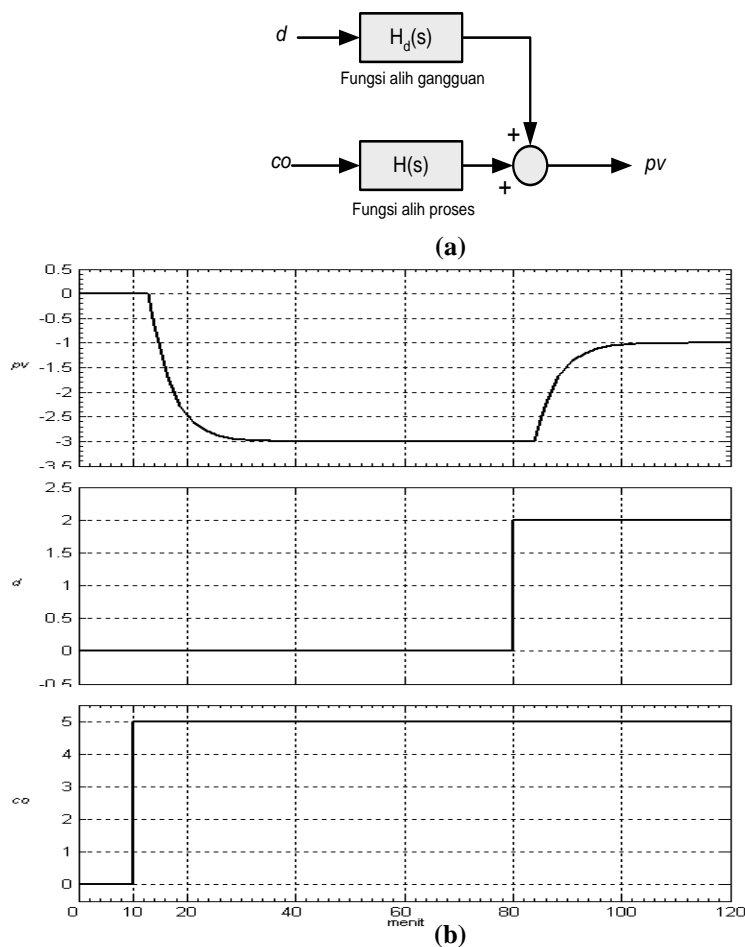
sehingga dengan demikian, untuk temperature akhir sebesar 180 °celcius, besar sinyal kontrol nominal yang diperlukan adalah:

$$\overline{CO}_1 = \frac{180 - 120}{2} + 40 = 70\%$$

#### Contoh 1.4

##### Analisis parameter proses FOPDT berdasarkan hasil uji eksperimen bump test (2)

Gambar 1.36 berikut memperlihatkan blok diagram sebuah proses beserta relasi input-outputnya.



**Gambar 1.36.** Blok diagram proses (a) dan relasi input-outputnya (b)

---

### Pertanyaan

Berdasarkan grafik relasi dalam gambar 1.36, carilah model matematis yang merepresentasikan hubungan input-output untuk sistem tersebut

### Penyelesaian

Dengan menganggap proses bersifat linear maka pengaruh kedua input (yaitu perubahan sinyal kontrol  $-co$  dan input gangguan  $-d$ ) terhadap deviasi output proses pada dasarnya dapat dicari berdasarkan pengaruh masing-masing input terhadap output proses tersebut:

#### - Pengaruh perubahan sinyal kontrol $-co$ terhadap deviasi output proses $-pv$

Model matematis perubahan output proses terhadap perubahan sinyal kontrol secara praktis pada dasarnya dapat dicari berdasarkan dinamika transien perubahan output yang terjadi sebagai akibat perubahan sinyal kontrol. Seperti nampak pada grafik, perubahan  $co$  untuk proses tersebut terjadi pada menit ke -10:

Untuk perubahan  $co$  sebesar 5 (satuan),  $pv$  mengalami perubahan sebesar 3(satuan), dengan deviasi bernilai negative ( $pv$  menuju pada keadaan *steady* yang lebih negative dari keadaan awalnya), sehingga dengan demikian nilai gain statis proses ( $K$ ) tersebut akan bernilai negative. Berdasarkan perumusan 1.1, besar gain statis tersebut adalah:

$$K = -\frac{3}{5} = -0.6$$

Sedangkan untuk nilai konstanta waktu- $T$  dan delay transportasi proses  $-L$ , berdasarkan pengamatan dari grafik, nilainya berturut-turut adalah 4 menit dan 3 menit. Sehingga dengan demikian model matematisnya dapat ditulis:

$$pv(s) = \frac{-0.6e^{-3s}}{4s + 1} co(s)$$

#### - Pengaruh input gangguan $-d$ terhadap deviasi output proses $-pv$

Seperti halnya pengaruh  $co$  terhadap  $pv$ , pengaruh  $d$  terhadap  $pv$  juga dapat dicari berdasarkan dinamika transien  $pv$  pada saat terjadinya perubahan besar variabel exogenous atau input gangguan tersebut. Dengan cara analisis yang sama seperti sebelumnya, pengaruh  $d$  terhadap  $pv$  akan memiliki model:

$$pv(s) = \frac{e^{-4s}}{4s + 1} d(s)$$

Dengan menggabungkan kedua pengaruh input tersebut, maka persamaan output proses didapat

$$pv(s) = \frac{-0.6e^{-3s}}{4s + 1} co(s) + \frac{e^{-4s}}{4s + 1} d(s)$$

### Contoh 1.5

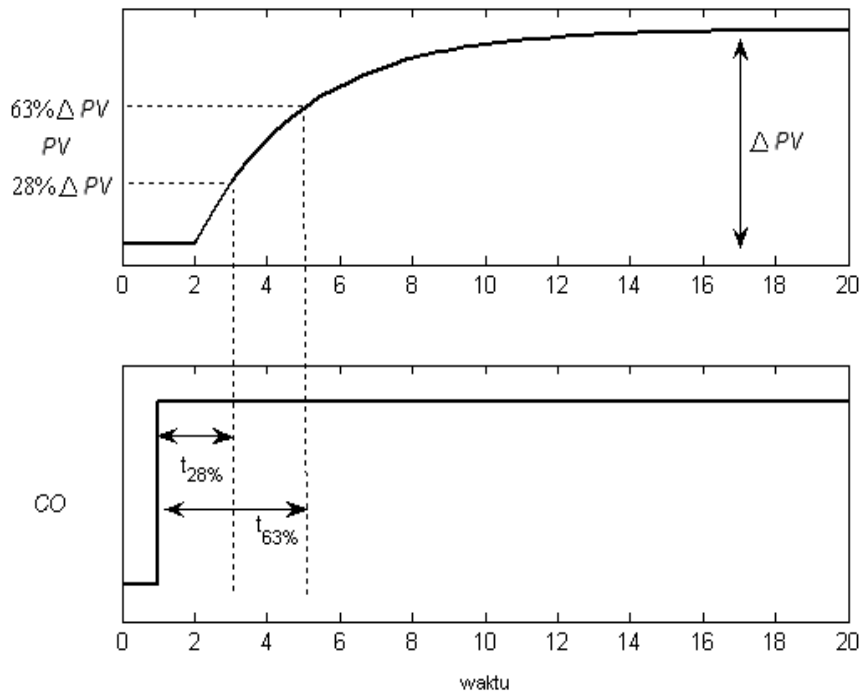
#### Analisis parameter proses FOPDT berdasarkan hasil uji eksperimen bump test (3)

Perumusan 1.15 dibawah memperlihatkan metode lain yang dapat digunakan untuk mencari nilai pendekatan konstanta waktu ( $T$ ) dan delay transportasi ( $L$ ) sebuah model proses FOPDT dari hasil uji Bump test.



$$T = 1.5(t_{63\%} - t_{28\%})$$
$$L = t_{63\%} - T$$
1.15

dalam hal ini  $t_{63\%}$  dan  $t_{28\%}$  berturut-turut adalah waktu yang diperlukan (dihitung sejak terjadi perubahan sinyal kontrol) sehingga output proses mencapai nilai 28% dan 63% dari keadaan akhirnya (lihat gambar 1.37).



**Gambar 1.37.** Perhitungan L dan T pada proses FOPDT berdasarkan hasil uji bump test

### Pertanyaan

Berdasarkan perumusan 1.15 tersebut, hitunglah nilai konstanta waktu ( $T$ ), dan delay transportasi ( $L$ ) untuk model *Heat Exchanger* contoh soal 1.3.

### Penyelesaian

Dengan mengacu grafik gambar 1.34 (atau gambar 1.35), nilai  $t_{28\%}$  dan  $t_{63\%}$  berturut-turut didapat sekitar 1.5 dan 2.5 menit, sehingga dengan demikian nilai  $L$  dan  $T$  dapat dihitung:

$$T = 1.5(t_{63\%} - t_{28\%}) = 1.5(2.5 - 1.5) = 1.5$$

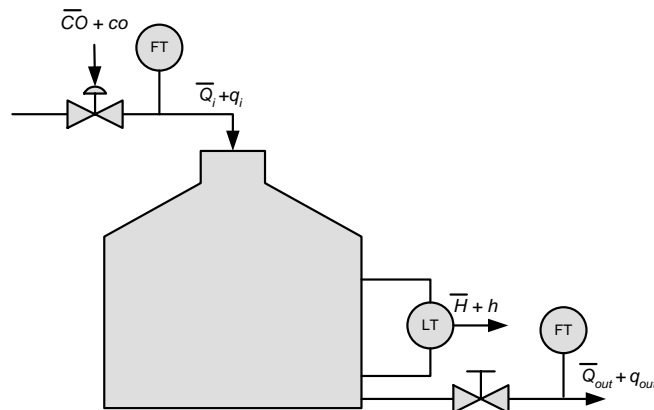
$$L = t_{63\%} - T = 2.5 - 1.5 = 1$$

Bandingkan hasil tersebut dengan jawaban contoh 1.3.(a)

### Contoh 1.6

#### Penurunan hukum fisika Model Kontrol proses level fluida pada sebuah tanki penampung

Tinjau sistem kontrol level fluida dengan penggerak *control valve* pada gambar 1.38 berikut ini



**Gambar 1.38.** Kontrol proses level fluida pada tangki penampung

### Pertanyaan

- Dengan menggunakan hukum Fisika, carilah model matematis antara deviasi level fluida ( $h$ ) terhadap perubahan sinyal kontrol ( $co$ ) disekitar nilai *steady* awalnya.
- Berdasarkan eksperimen pada titik kerjanya, perubahan ketinggian sebesar 0.1 m (yang disebabkan oleh perubahan debit fluida input) pada tangki ternyata menyebabkan perubahan debit output sebesar 0.05 m<sup>3</sup>/s, jika diketahui luas penampang tangki  $A = 1$  m<sup>2</sup> dan proporsionalitas antara perubahan debit masukan terhadap perubahan sinyal kontrol bernilai 0.25. Carilah model yang merelasikan hubungan perubahan ketinggian level fluida terhadap perubahan sinyal kontrol untuk proses tersebut

### Penyelesaian

- Dengan menganggap posisi *control valve* berdekatan dengan tangki penampung, maka model matematis perubahan ketinggian ( $h$ ) terhadap perubahan sinyal kontrol ( $co$ ) disekitar nilai *steady* nominalnya dapat dicari dengan menggunakan Hukum kesetimbangan massa berikut:

Laju akumulasi massa pada tangki = laju massa input – laju massa output

Atau dapat ditulis secara matematis :

$$\frac{d}{dt}(\rho Ah(t)) = \rho q_{in} - \rho q_{out} \quad (1.16)$$

dengan:

- $q_{in}$  = perubahan laju aliran fluida input disekitar nilai nominalnya (m<sup>3</sup>/det)
- $q_{out}$  = perubahan laju aliran fluida output disekitar nilai nominalnya(m<sup>3</sup>/det)
- $\rho$  = densitas fluida(kg/m<sup>3</sup>)
- $A$  = luas penampang tangki (m<sup>2</sup>)
- $h$  = Perubahan ketinggian fluida dalam tangki (m)

Pada sistem tangki penampung tersebut, perubahan debit aliran fluida output pada dasarnya akan berbanding lurus dengan perubahan ketinggian fluida pada tangki ( $h$ ) :

$$q_{out} = K_{out} h(t) \quad (1.17)$$

dengan :

---

$K_{out}$  = konstanta proporsionalitas yang harganya tergantung bukaan valve output  
Jika *control* valve yang digunakan dianggap bertipe ATO, maka perubahan debit fluida input akan proporsional terhadap besar perubahan sinyal kontrol (*co*) penggerak *valve*:

$$q_{in} = K_{in} co \quad (1.18)$$

dengan mensubstitusikan persamaan (1.17) dan (1.18) pada persamaan (1.16) serta dengan melakukan sedikit langkah penyederhanaan, maka akan diperoleh persamaan diferensial linear yang ekuivalen dengan bentuk persamaan (1.2)

$$\frac{A}{K_{out}} \frac{dh(t)}{dt} = -h(t) + \frac{K_{in}}{K_{out}} co(t) \quad (1.19)$$

Atau dapat ditulis dalam bentuk umum:

$$T \frac{dh(t)}{dt} = -h(t) + K.co(t) \quad (1.20)$$

Sehingga dengan demikian, fungsi alih antara perubahan level fluida terhadap perubahan sinyal kontrol disekitar titik kerja nominalnya adalah:

$$\frac{h(s)}{co(s)} = \frac{K}{Ts + 1} \quad (1.21)$$

dengan :

$$K = \frac{K_{in}}{K_{out}} \text{ (gain statis proses)}$$

$$T = \frac{A}{K_{out}} \text{ (Konstanta waktu proses)}$$

**b. Diketahui :**

$$A = 1 \text{ m}^2$$

$$K_{out} = \frac{0.05}{0.1} = 0.5 \text{ m}^2/\text{s}$$

$$K_{in} = 0.25 \text{ m}^3/\text{s}\%$$

$$K = \frac{K_{in}}{K_{out}} = \frac{0.25}{0.5} = 0.5$$

$$T = \frac{A}{K_{out}} = \frac{1}{0.5} = 2$$

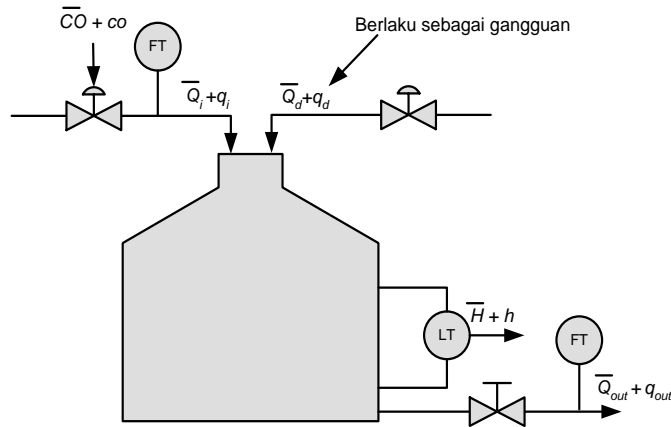
Sehingga berdasarkan persamaan (1.21) didapat:

$$H(s) = \frac{h(s)}{co(s)} = \frac{0.5}{2s + 1}$$

### Contoh 1.7

#### Model Kontrol proses level fluida dengan input gangguan

Tinjau sistem kontrol level fluida pada gambar 1.39 berikut:



Gambar 1.39. Sistem Kontrol level fluida dengan gangguan

#### Pertanyaan

Dengan menggunakan hukum kesetimbangan massa disekitar daerah kerjanya, carilah model matematis (fungsi alih) antara output proses (deviasi level fluida dalam tanki) terhadap perubahan sinyal kontrol ( $co$ ) serta input gangguan (perubahan debit fluida input -  $q_d$ ) yang terjadi dalam proses tersebut. Untuk menyederhanakan persoalan anggap kedua jenis fluida yang masuk ke dalam tangki memiliki densitas yang sama.

#### Penyelesaian

Berdasarkan Hukum Kesetimbangan massa :

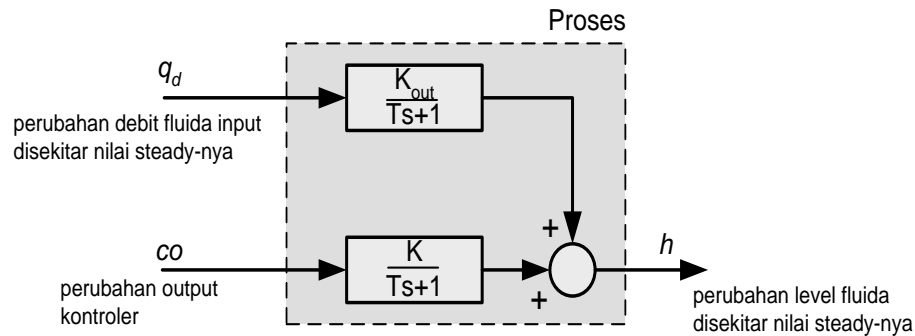
$$\frac{d}{dt}(\rho Ah(t)) = \rho q_{in} + \rho q_d - \rho q_{out} \quad (1.22)$$

Dengan cara yang identik seperti pada contoh 1.5, dapat dibuktikan hasil penyelesaian persamaan 1.20 diatas adalah:

$$h(s) = \frac{K}{Ts + 1} co(s) + \frac{K_{out}}{Ts + 1} q_d(s) \quad (1.23)$$

(dalam hal ini parameter  $K$ ,  $K_{out}$  dan  $T$  penjelasannya sama dengan contoh 1.6).

Dalam bentuk diagram blok-nya, persamaan (1.23) tersebut akan nampak seperti terlihat pada gambar (1.40) dibawah ini:

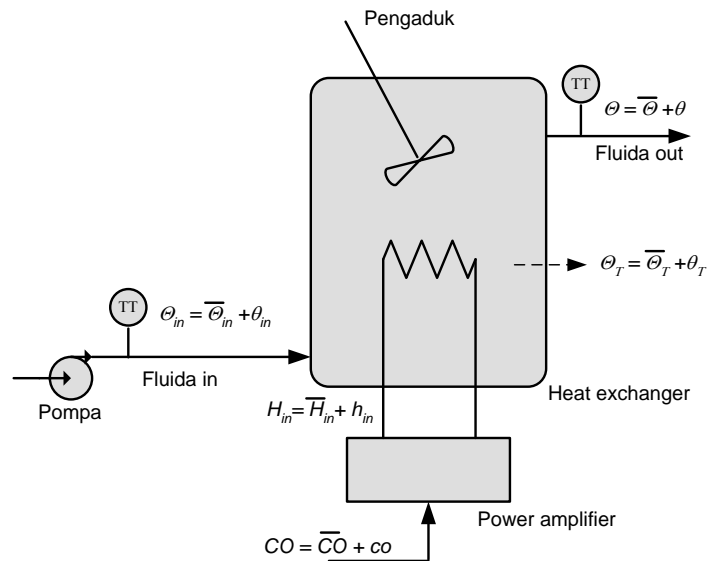


**Gambar 1.40.** Diagram blok sistem kontrol level fluida untuk gambar 1.39

### Contoh 1.8

#### Penurunan hukum fisika Model Kontrol proses Temperature pada sistem heat exchanger

Tinjau sistem kontrol proses pemanasan fluida seperti nampak pada gambar 1.41 berikut:



**Gambar 1.41.** Sistem Kontrol Heat Exchanger

#### Pertanyaan

- Carilah model matematis antara deviasi temperature fluida output terhadap perubahan sinyal kontrol dan perubahan temperature fluida input disekitar nilai *steady*-nya.
- Gambarkan blok diagram untuk model proses tersebut
- Berdasarkan jawaban (a), carilah hubungan perubahan temperature fluida output terhadap besar perubahan sinyal output kontroler dan perubahan temperature fluida input untuk laju aliran fluida ( $w$ ) = 20 kg/menit.

Dimana parameter-parameter proses tersebut diketahui:

$$V \text{ (Volume tangki)} = 0.2 \text{ m}^3$$

$$\rho \text{ (densitas fluida)} = 1000 \text{ kg/m}^3$$

---

$c$  (Kalor jenis fluida) = 4000 J/(Kg.°C)

$K_{ps}$  (konstanta kesebandingan antara perubahan panas dengan perubahan sinyal kontrol) = 800 W/% (800.60 J/menit/% = 48000 J/menit/%)

$K_L$  = 50 (Kg)

- d. Jika model proses sistem *Heat exchanger* dianggap linear untuk keseluruhan daerah kerja, serta sistem terisolasi dengan lingkungan luar, carilah:
- Nilai temperature fluida output pada keadaan *steady*-nya untuk besar sinyal kontrol nominal 0%
  - Nilai temperature fluida output pada keadaan *steady*-nya untuk besar sinyal kontrol nominal sebesar 100% (sinyal kontrol maksimum) dengan temperature fluida input bernilai 25 °C
  - Besar sinyal kontrol yang diperlukan untuk membawa temperature fluida di sisi output sehingga menetap pada nilai 65° C (anggap temperature fluida input bernilai 20 °C)

### Penyelesaian

- a. Dengan menggunakan hukum kesetimbangan energi disekitar daerah kerjanya, maka model matematis antara output proses (deviasi temperature output) terhadap perubahan input yang terjadi dalam proses tersebut dapat dicari sebagai berikut (Untuk menyederhanakan persoalan, anggap *heat exchanger* tersebut terisolasi penuh dengan lingkungan):

$$\frac{d}{dt}(V\rho c\theta_T(t)) = cw\theta_{in} - wc\theta_T + h_{in} \quad (1.24)$$

dengan:

- $V$  : Volume tangki (konstan) - m<sup>3</sup>  
 $w$  : laju aliran fluida dalam keadaan steady -kg/menit  
 $\rho$  : densitas fluida (konstan) – kg/m<sup>3</sup>  
 $c$  : Kalor jenis fluida (konstan) – J/(Kg.°C)  
 $\theta_{in}$  : perubahan temperature fluida input disekitar nilai *steady* awalnya (°C)  
 $\theta_T$  : perubahan temperature fluida dalam tangki disekitar nilai *steady* awalnya (°C)  
 $h_{in}$  : perubahan laju kalor masukan disekitar nilai *steady* awalnya (J/menit)

Jika penguat daya yang digunakan bersifat linear, maka perubahan laju kalor masukan akan sebanding dengan besar perubahan sinyal kontrol – co (%):

$$h_{in} = K_{ps}co \quad (1.25)$$

dengan mensubstitusikan persamaan (1.25) pada persamaan (1.24) serta dengan melakukan sedikit langkah penyederhanaan, maka akan diperoleh persamaan diferensial berikut:

$$\frac{V\rho}{w} \frac{d}{dt}(\theta_T(t)) = \theta_{in} - \theta_T + \frac{K_{ps}co}{cw} \quad (1.26)$$

atau dapat ditulis:

$$T \frac{d}{dt}(\theta_T(t)) = \theta_{in} - \theta_T + Kco \quad (1.27)$$

dalam hal ini:

$$\text{konstanta waktu proses : } T = \frac{V\rho}{w}$$

$$\text{gain statis proses: } K = \frac{K_{ps}}{cw}$$

Dengan mencari transformasi laplace untuk ke dua ruas persamaan 1.27 diatas, maka akan diperoleh persamaan 1.28 berikut:

$$\theta_T(s) = \frac{K}{Ts+1} co(s) + \frac{1}{Ts+1} \theta_{in}(s) \quad (1.28)$$

Jika ketidak homogenan kondisi di dalam *heat exchanger* ikut diperhitungkan, maka temperature yang terbaca di sisi outlet *heat exchanger* tersebut secara praktis akan mengalami keterlambatan (*delay*) yang nilainya akan berbanding terbalik dengan besar laju fluida, atau secara matematis dapat ditulis seperti terlihat pada persamaan 1.29.

$$\theta(t) = \theta_T(t - L) \quad (1.29)$$

dengan :

$\theta$  : perubahan temperature fluida output disekitar nilai *steady* awalnya (°C)

$$L = \frac{K_L}{w}; \quad K_L = \text{konstanta keterlambatan}$$

dalam kawasan laplace persamaan 1.29 diatas dapat direpresentasikan seperti nampak pada persamaan 1.30 dibawah:

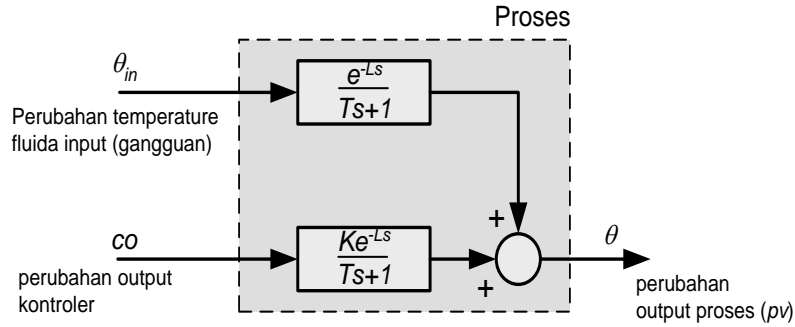
$$\theta(s) = e^{-Ls} \theta_T(s) \quad (1.30)$$

dengan mensubsitusikan persamaan 1.28 ke dalam persamaan 1.30 maka didapat:

$$\theta(s) = \frac{Ke^{-Ls}}{Ts+1} co(s) + \frac{e^{-Ls}}{Ts+1} \theta_{in}(s) \quad (1.31)$$

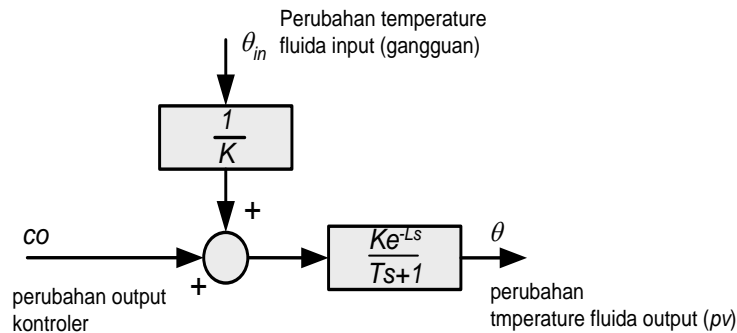
Berdasarkan persamaan 1.31 nampak bahwa dinamika proses (nilai parameter proses) akan sangat tergantung pada titik kerja (debit fluida) yang temperature-nya akan diregulasi: Semakin besar debit fluida maka proses semakin tidak sensitif (gain statis  $-K$  semakin kecil), kecepatan respon semakin cepat (konstanta waktu- $T$  semakin kecil) dan Keterlambatan transportasi- $L$  semakin kecil (perhatikan penjelasan pada persamaan 1.27 dan 1.29)

- b. Dalam bentuk diagram bloknya, persamaan 1.31 diatas akan nampak seperti pada gambar 1.42 berikut:



**Gambar 1.42.** Diagram blok Sistem Kontrol *Heat Exchanger*

Diagram blok pada gambar 1.42 diatas pada dasarnya dapat disederhanakan seperti nampak pada gambar 1.43.



**Gambar 1.43.** Diagram blok Sistem Kontrol *Heat Exchanger* alternative

- c. Berdasarkan penurunan hukum Fisika pada jawaban (a), maka parameter- parameter proses heat exchanger dapat dihitung sebagai berikut:

$$T = \frac{V\rho}{w} = \frac{0.2 \times 1000}{20} = 10 \text{ (menit)}$$

$$K = \frac{K_{ps}}{cw} = \frac{48000}{4000 \times 20} = 0.6 \text{ (}^\circ\text{C/\%)}$$

$$L = \frac{K_L}{w} = \frac{50}{20} = 2.5 \text{ (menit)}$$

Dengan mensubstitusikan parameter diatas pada persamaan (1.31), maka model proses heat exchanger disekitar daerah kerjanya tersebut adalah:

$$\theta(s) = \frac{0.6e^{-2.5s}}{10s+1} co(s) + \frac{e^{-2.5s}}{10s+1} \theta_i(s) \quad (1.32)$$

- d. Jika model proses sistem *Heat exchanger* kita anggap linear, serta sistem tersebut benar-benar terisolasi dengan lingkungan luar (sehingga tidak terjadi pertukaran panas dengan lingkungannya) maka variabel input-output model proses pada persamaan 1.32 pada dasarnya dapat kita tulis dalam bentuk variabel nominalnya (variabel yang terukur sensor --lihat persamaan 1.33 dan diagram prosesnya pada gambar 1.44).



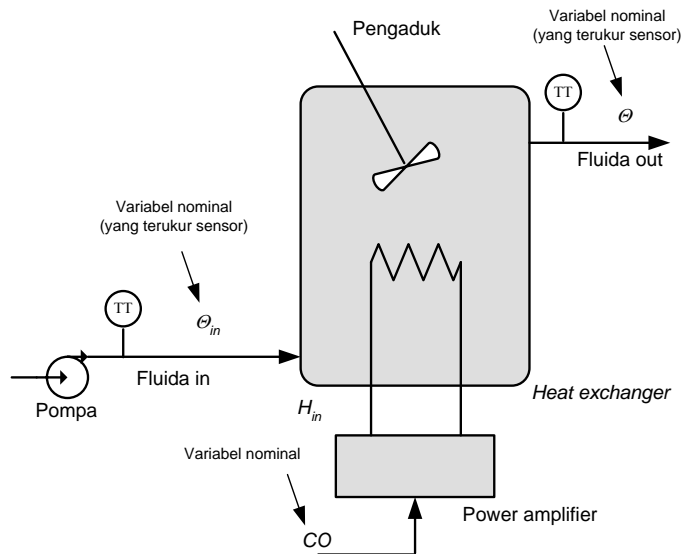
$$\Theta(s) = \frac{0.6e^{-2.5s}}{10s+1}CO(s) + \frac{e^{-2.5s}}{10s+1}\Theta_{in}(s) \quad (1.33)$$

dengan :

$\Theta$  = Temperature fluida output nominal

$CO$  = Output Kontroler nominal

$\Theta_{in}$  = Temperature fluida input nominal



**Gambar 1.44.** Model Heat exchanger dianggap benar-benar linear

Untuk mengetahui harga akhir (atau nilai output proses dalam keadaan *steady*-nya), kita dapat menggunakan relasi pada persamaan 1.2 (hanya saja karena prosesnya linear serta semua variabel input telah diakomodasi dalam persamaan, maka variabel deviasi pada persamaan tersebut, pada dasarnya langsung dapat diganti dengan variabel nominalnya):

Hubungan sinyal kontrol dan temperature fluida input dengan temperature output pada keadaan *steady*:

$$\Theta = K.CO + K_d\Theta_{in}$$

Berdasarkan model, diketahui bahwa gain statis proses --  $K = 0.6$  dan gain statis gangguan --  $K_e = 1$ . Dengan demikian:

- Nilai temperature fluida output pada keadaan *steady* untuk besar sinyal kontrol nominal 0% dapat dicari

$$\Theta = 0.6x0 + \Theta_{in}$$

atau:

$$\Theta = \Theta_{in}$$

Berdasarkan persamaan terakhir nampak bahwa jika besar sinyal kontrol sama dengan 0%, maka temperature fluida output pada dasarnya akan selalu sama dengan temperature fluida input-nya.

- Nilai temperature fluida output pada keadaan *steady* untuk besar sinyal kontrol nominal sebesar 100% (sinyal kontrol maksimum) dan temperature fluida input 25 °C dapat dicari

$$\Theta = K.CO + K_d.\Theta_{in}$$

$$\Theta = 0.6 \times 100 + 1 \times 25$$

$$\Theta = 85^\circ \text{ celcius}$$

- Besar sinyal kontrol yang diperlukan untuk membawa temperature fluida di sisi output sehingga menetap pada nilai 65° C (anggap temperature fluida input bernilai 20 °C) dapat dicari

$$\Theta = K.CO + K_d.\Theta_{in}$$

$$65 = 0.6 \times CO + 1 \times 20$$

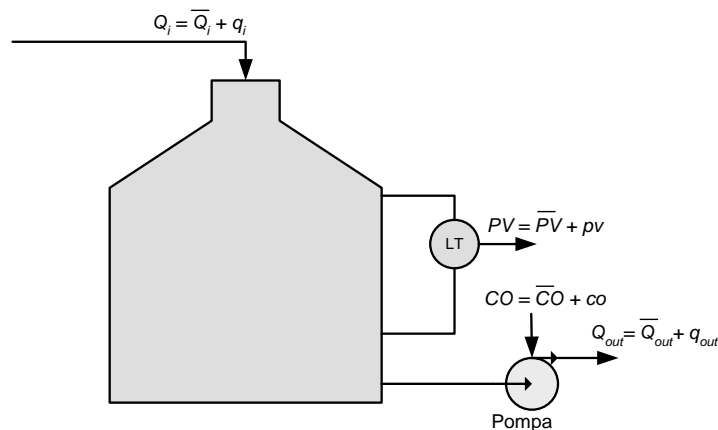
$$CO = \frac{65 - 20}{0.6}$$

$$CO = 75\%$$

### Contoh 1.9

#### Penurunan hukum fisika Model Kontrol proses level fluida dengan penggerak pompa penghisap

Tinjau diagram Proses pada gambar 1.45 berikut:



Gambar 1.45. Sistem Kontrol level fluida dengan penggerak pompa

### Pertanyaan

- a. Carilah model matematis antara deviasi level fluida dalam tangki terhadap perubahan sinyal kontrol pompa dan debit fluida input disekitar nilai kesetimbangannya.

- b. Berdasarkan jawaban (a), Carilah model matematis untuk sistem kontrol tersebut jika diketahui luas penampang tangki ( $A$ ) dan konstanta kesebandingan antara perubahan debit output terhadap perubahan output kontroler ( $K_{out}$ ) berturut-turut bernilai  $2 \text{ m}^2$  dan  $0.1 \text{ m}^3/(\text{menit.}\%)$ .

### Penyelesaian

- a. Berdasarkan hukum kesetimbangan massa:

Laju akumulasi massa pada tangki = laju massa input – laju massa output

$$\frac{d}{dt}(\rho Ah(t)) = \rho q_{in} - \rho q_{out} \quad (1.34)$$

dengan:

- $q_{in}$  = perubahan laju aliran fluida input disekitar nilai nominalnya ( $\text{m}^3/\text{det}$ )  
 $q_{out}$  = perubahan laju aliran fluida output disekitar nilai nominalnya ( $\text{m}^3/\text{det}$ )  
 $\rho$  = densitas fluida ( $\text{kg}/\text{m}^3$ )  
 $A$  = luas penampang tangki ( $\text{m}^2$ )  
 $h$  = Perubahan ketinggian fluida dalam tangki (m)

Pada sistem tangki penampung tersebut, perubahan debit aliran fluida output pada dasarnya akan berbanding lurus dengan perubahan sinyal kontrol pompa ( $co$ ) :

$$q_{out} = K_{out} co \quad (1.35)$$

Dengan mensubsitusikan persamaan (1.35) kedalam persamaan (1.34) dan dengan sedikit langkah penyederhanaan maka didapat:

$$A \frac{d}{dt} h(t) = q_{in} - K_{out} co \quad (1.36)$$

Jika ditulis dalam kawasan Laplace maka persamaan (1.36) diatas akan menjadi:

$$h(s) = \frac{1}{As} q_i(s) - \frac{K_{out}}{As} co(s) \quad (1.37)$$

dengan membandingkan persamaan (1.37) dengan persamaan (1.13) --bentuk umum persamaan model IPDT, nampak bahwa :

$$\text{Gain integratif proses : } K^* = -\frac{K_{out}}{A} \quad (1.38)$$

$$\text{Gain integratif gangguan: } K_d^* = \frac{1}{A} \quad (1.39)$$

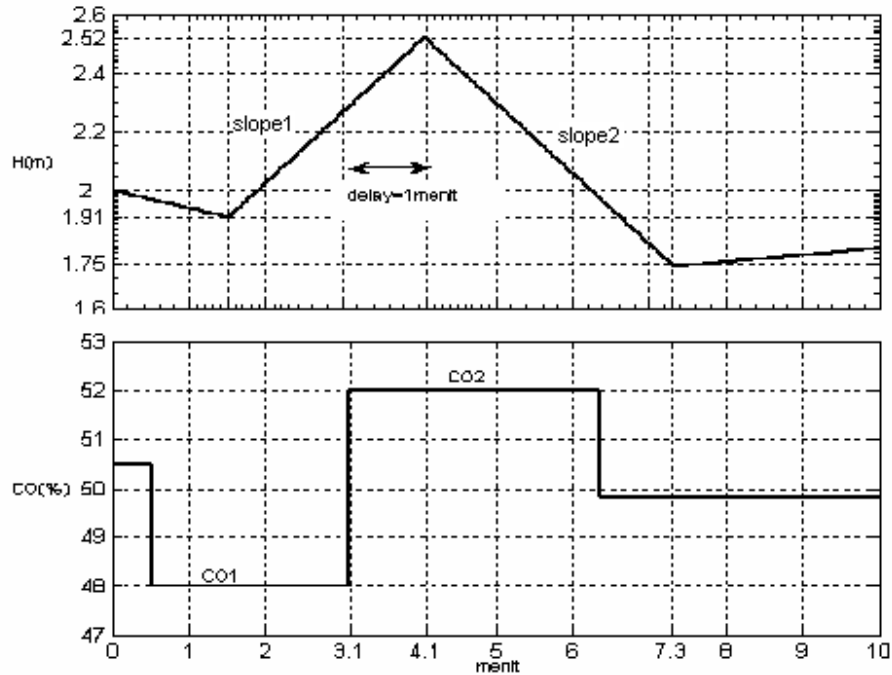
- b. Dengan mensubsitusikan nilai parameter  $A$  dan  $K_{out}$  pada persamaan 1.37, maka didapat:

$$h(s) = \frac{1}{2s} q_i(s) - \frac{0.05}{s} co(s)$$

### Contoh 1.10

#### Studi kasus Sistem linear untuk kontrol level

Gambar 1.46 dibawah ini memperlihatkan hubungan antara besar sinyal kontrol pompa ( $CO$ ) terhadap level fluida pada sebuah sistem kontrol level dalam tanki penampung (anggap diameter penampang tangki --  $A = 2 \text{ m}^2$ ).



**Gambar 1.46.** Hubungan input-output pada sistem kontrol level dengan penggerak sebuah pompa

#### Pertanyaan:

- Carilah model matematis proses berdasarkan grafik relasi tersebut
- Berapakah konstanta kesebandingan antara perubahan debit output terhadap perubahan output kontroler penggerak pompa ( $K_{out}$ )
- Jika sistem kontrol tersebut kita anggap linear untuk seluruh daerah kerjanya (jangkauan sinyal kontrol --  $CO$  antara 0% - 100%), serta dianggap tidak ada delay transportasi pada sisi fluida inputnya, representasikanlah model matematis lengkap untuk sistem tersebut.
- Jika debit fluida input bernilai  $12 \text{ m}^3/\text{menit}$ , berapakah besar sinyal kontrol - $CO$  yang harus diseting sedemikian sehingga level fluida pada tanki akan menetap (stabil) pada nilai tertentu.

#### Penyelesaian

- Berdasarkan persamaan 1.14, gain integrative proses dapat dicari:

$$K^* = \frac{\Delta \text{slope}}{\Delta CO} = \frac{\text{slope}_2 - \text{slope}_1}{CO_2 - CO_1}$$

Dengan melihat grafik pada gambar 1.46, didapat:

$$slope_2 = \frac{1.75 - 2.52}{7.3 - 4.1} = -0.24$$

$$slope_1 = \frac{2.52 - 1.91}{4.1 - 1.5} = 0.235$$

$$\Delta CO = CO_2 - CO_1 = 52 - 48 = 4$$

sehingga dengan demikian nilai gain integrative prosesnya :

$$K^* = \frac{\Delta slope}{\Delta CO} = \frac{-0.24 - 0.23}{4} = -0.12$$

Sedangkan besar nilai delay transportasi proses -- $L = 1$  menit (lihat gambar 1.46)

Dengan demikian, relasi antara perubahan level terhadap perubahan sinyal kontrol pompa dapat ditulis:

$$h(s) = -\frac{0.12e^{-s}}{s}co(s)$$

**b.** Berdasarkan persamaan (1.38) diketahui bahwa :

$$K^* = -\frac{K_{out}}{A}$$

atau

$$K_{out} = -K^* . A$$

$$K_{out} = -(-0.12) \times 2 = 0.24 \text{ m}^3/(\text{menit.}\%)$$

**c.** Jika proses bersifat linear untuk jangkauan sinyal kontrol -- $CO$  antara 0%-100%, maka berdasarkan bentuk umum model proses IPDT :

$$PV(s) = \frac{K^*}{s} e^{-sL} CO(s) + \frac{K_d^*}{s} e^{-sL_d} Q_i(s)$$

dalam hal ini  $K^* = -\frac{K_{out}}{A} = -0.12$

$$K_d^* = \frac{1}{A} = 0.5 \text{ (lihat relasi 1.37)}$$

sehingga dengan demikian:

$$PV(s) = -\frac{0.12}{s} e^{-s} . CO(s) + \frac{0.5}{s} Q_i(s)$$

keterangan: karena sistemnya dianggap linear maka semua variable input dan output pada proses diatas dapat kita tulis dalam bentuk variable nominalnya

- d. Untuk sistem kontrol pompa tersebut, level fluida akan menetap atau stabil pada sebuah nilai tentunya jika debit fluida output ( $Q_{out}$ ) = Debit fluida input ( $Q_i$ ). Jika diketahui  $Q_i = 12 \text{ m}^3/\text{menit}$ , maka besar sinyal kontrol yang diperlukan agar debit fluida juga sama dengan  $12 \text{ m}^3/\text{menit}$ , pada dasarnya dapat dicari dari relasi proporsionalitas antara  $Q_{out}$  dan  $CO$  --lihat persamaan 1.35

$$Q_{out} = K_{out} \cdot CO$$

$$CO = \frac{Q_{out}}{K_{out}}$$

berdasarkan jawaban (b) diketahui  $K_{out} = 0.24$ , sehingga dengan demikian

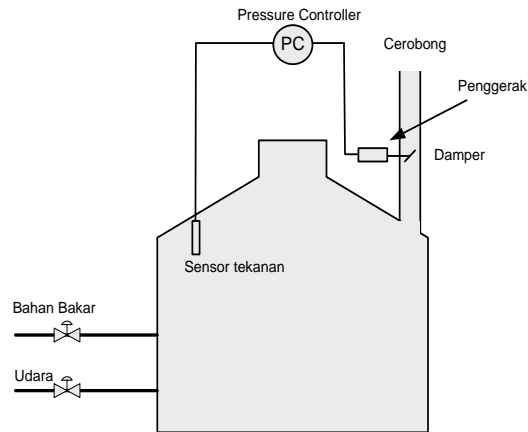
$$CO = \frac{12}{0.24} = 50\%$$

## 1.7 Ringkasan

- Kontrol proses industri dalam banyak kasus umumnya akan menunjukkan salah satu dari dua model berikut:
  - (1) Model proses yang bersifat stabil (*Self Regulating Process*). Untuk kepentingan perancangan dan tuning parameter PID praktis, model proses ini secara umum dapat didekati oleh sebuah model matematis yang dikenal dengan nama model **FOPDT** (*First Order Plus Dead Time*) yang hanya dicirikan oleh tiga buah parameter berikut: Delay transportasi (**L**), Konstanta waktu proses (**T**) dan Gain statis proses (**K**).
  - (2) Model proses yang tidak stabil (*Non Self Regulating Process*). Salah satu yang umum dijumpai di industri adalah model **IPDT(Integrating Plus Dead Time)** yang hanya dicirikan oleh dua buah parameter berikut: Delay transportasi (**L**) dan Gain integratif (**K\***).
- Parameter –parameter sebuah kontrol proses (baik model FOPDT maupun IPDT secara praktis pada dasarnya dapat diidentifikasi lewat eksperimen yang dikenal dengan nama *Bump Test Experiment*, yaitu dengan cara mengubah nilai output kontroler secara mendadak oleh operator. Output proses kemudian direkam dan hasilnya dianalisis.
- Model proses pada kenyataannya hanya valid pada daerah kerja yang terbatas (yaitu disekitar titik kerjanya), hal ini muncul dari kenyataan bahwa dalam daerah kerja yang luas, proses itu sendiri bersifat nonlinear. Sehingga dengan demikian, variabel-variabel yang muncul dalam persamaan matematis (baik dalam bentuk persamaan diferensial maupun dalam bentuk fungsi alih) harus ditulis dalam bentuk variabel deviasi disekitar titik kerja proses tersebut (Tetapi walaupun demikian jika nonlinearitas proses tidak terlalu signifikan, dan semua variable input telah diakomodasi pada model matematis proses tersebut, maka untuk memudahkan analisis, variabel input-output proses tersebut dapat langsung kita representasikan dalam bentuk variabel nominalnya).
- Penggunaan perangkat bantu simulasi komputer seperti Simulink pada Matlab secara praktis akan sangat membantu mempercepat pemahaman dinamika sebuah proses yang akan/sedang dilakukan aksi pengontrolan.

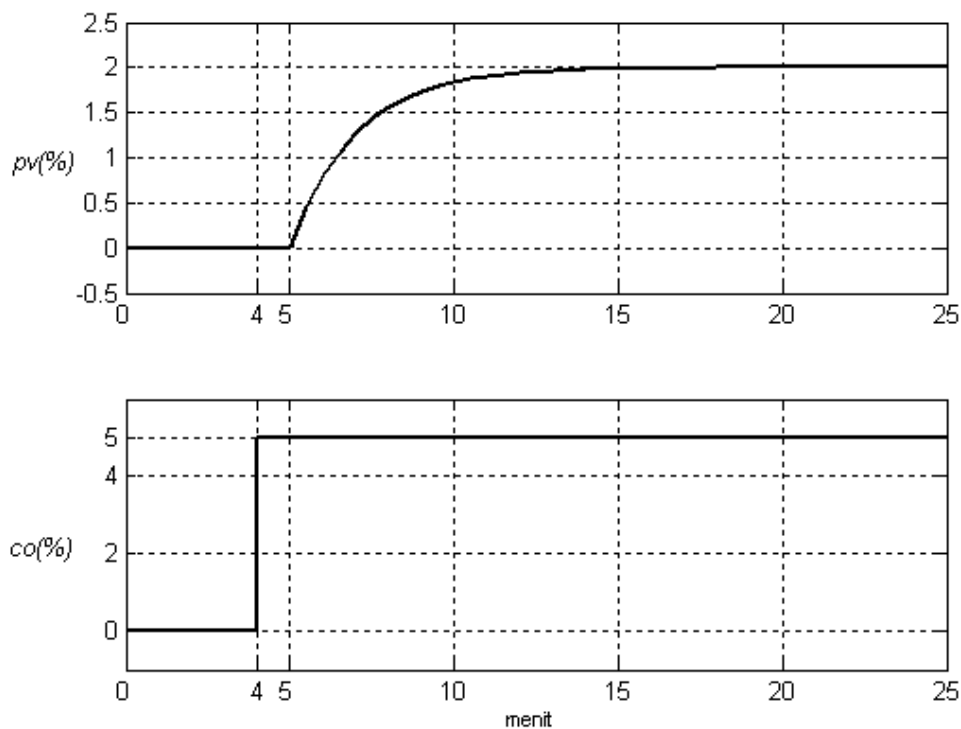
## 1.8 Soal Latihan

1.1. Jelaskan prinsip kerja dari sistem pengontrolan tekanan pada sebuah reaktor seperti nampak pada gambar 1.44 berikut (dan gambarkan pula diagram blok-nya).



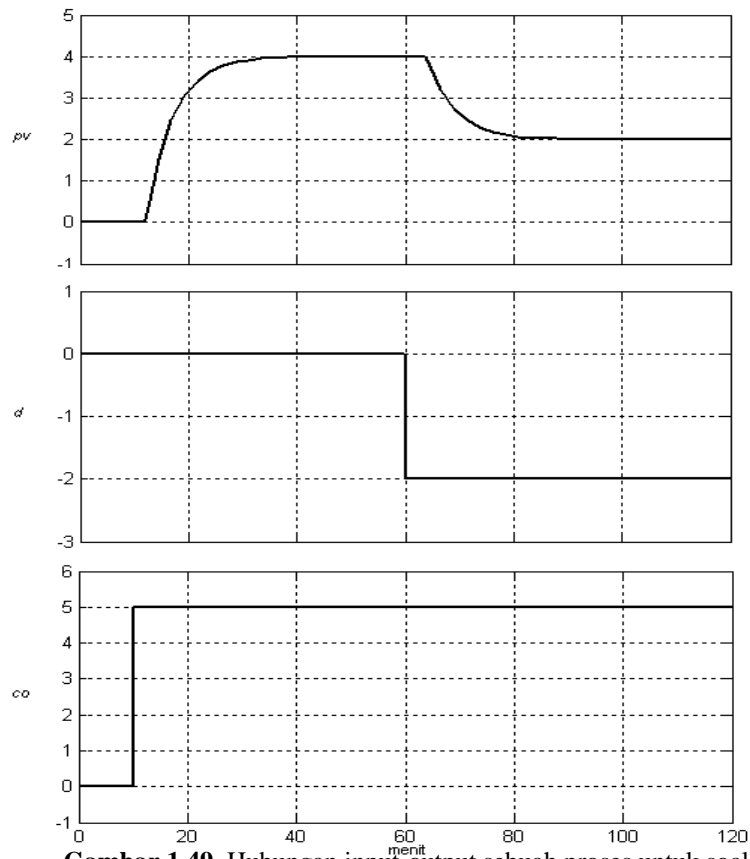
Gambar 1.47. Diagram instrumentasi proses untuk soal 1.1

1.2. Berdasarkan grafik pada gambar 1.48 carilah konstanta waktu proses berdasarkan slope maksimum respon transien.



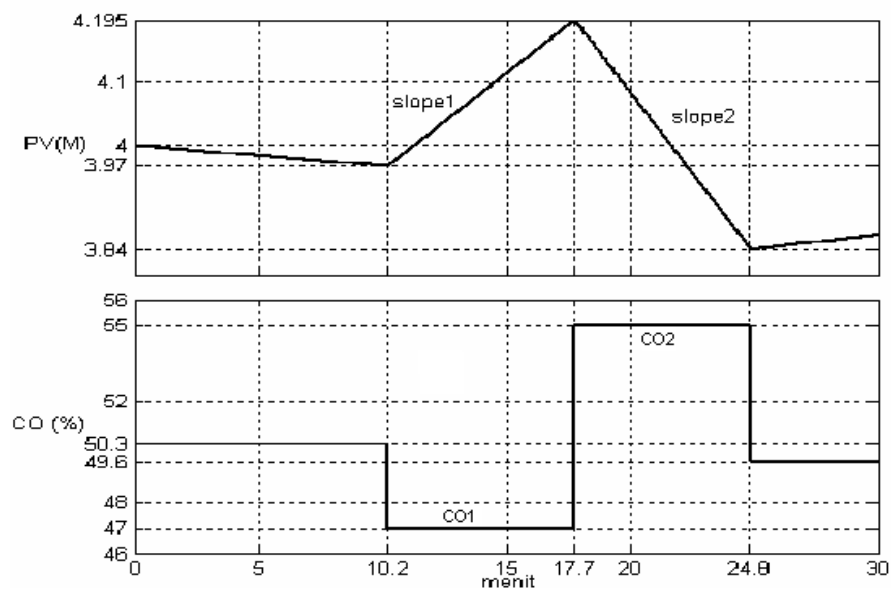
Gambar 1.48. Diagram instrumentasi proses untuk soal 1.2

1.3. Carilah model matematis sebuah proses yang memiliki hubungan input-output seperti nampak pada gambar 1.49 berikut



**Gambar 1.49.** Hubungan input-output sebuah proses untuk soal 1.3

**1.4.** Gambar 1.50 dibawah ini memperlihatkan hasil eksperimen bump test pada sistem kontrol level dengan penggerak pompa

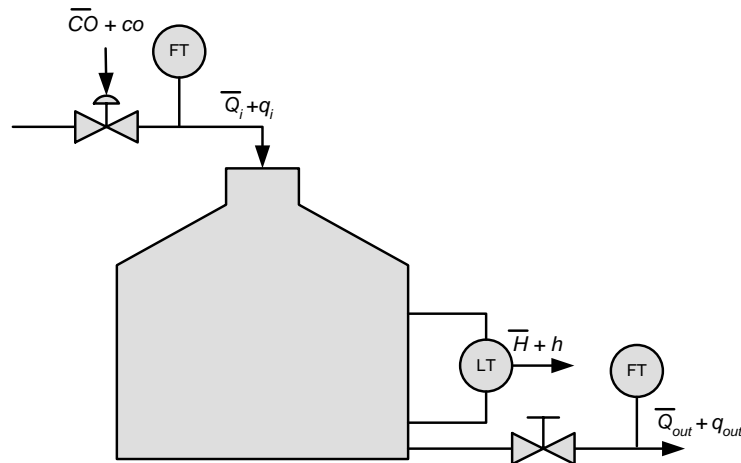


**Gambar 1.50.** Hubungan input-output pada sistem kontrol level dengan penggerak sebuah pompa



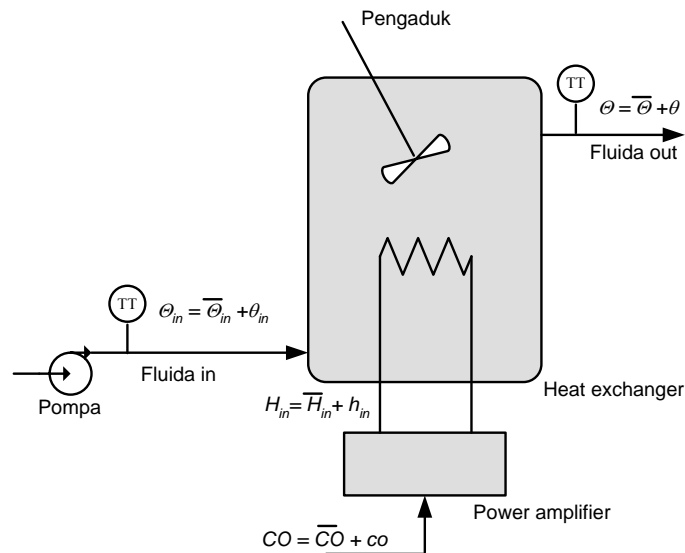
Berdasarkan grafik diatas, carilah gain integrative sekaligus juga representasi model matematis IPDT-nya.

**1.5.** Dengan menggunakan hukum kesetimbangan massa disekitar keadaan *steady*-nya, carilah relasi perubahan debit keluaran ( $q_{out}$ ) terhadap perubahan sinyal control ( $co$ ) untuk proses dibawah ini



**Gambar 1.51.** Diagram instrumentasi proses untuk soal 1.5

**1.6.** Tinjau sistem heat exchanger pada gambar 1.52 berikut



**Gambar 1.52.** Diagram instrumentasi proses untuk soal 1.6

Carilah dinamika proses atau persamaan (dalam kawasan laplace) perubahan temperature keluaran terhadap perubahan sinyal control dan deviasi temperature masukan disekitar titik operasi sistem kontrol diatas untuk nilai parameter-parameter proses berikut:

$$V \text{ (Volume tangki)} = 0.3 \text{ m}^3$$

$w$  (laju aliran fluida dalam keadaan steady) = 60 kg/menit

$\rho$  (densitas fluida) = 1000 kg/m<sup>3</sup>

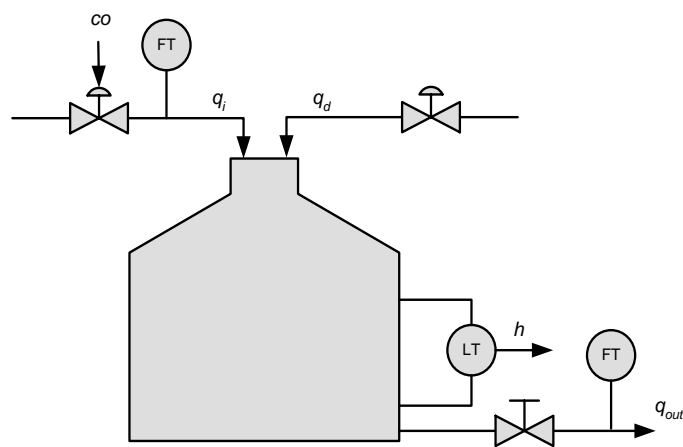
$c$  (Kalor jenis fluida) = 4000 J/(Kg.°C)

$K_{ps}$  (konstanta kesebandingan antara perubahan panas dengan perubahan sinyal kontrol)  
= 800 W/%

$K_L$  = 40 Kg

**1.7.** Dengan menggunakan simulink, simulasikan dinamika proses untuk soal 1.6 diatas.

**1.8.** Dengan menggunakan simulink, simulasikanlah dinamika kontrol proses gambar 1.53 berikut pada keadaan steady state-nya:



**Gambar 1.53.** Diagram instrumentasi proses untuk soal 1.8

Untuk kepentingan simulasi, anggap proses diatas memiliki parameter berikut:

$$A = 2 \text{ m}^2$$

$$K_{out} = 0.7 \text{ m}^2/\text{s}$$

$$K_{in} = 0.2 \text{ m}^3/\text{s}\%$$

**1.9** Dengan menggunakan simulink, simulasikanlah model-model proses berikut:

$$\text{a. } pv(s) = \frac{0.5}{2s+1} e^{-1.5s} .co(s)$$

$$\text{b. } pv(s) = \frac{-0.5}{2s+1} e^{-1.5s} .co(s)$$

$$\text{c. } pv(s) = \frac{0.5}{2s+1} e^{-1.5s} .co(s) + \frac{1}{2s+1} e^{-0.5s} d(s)$$

$$\text{d. } pv(s) = \frac{-0.5}{s} e^{-1.2s} .co(s)$$

---

$$\text{e. } pv(s) = \frac{-0.5}{s} e^{-1.2s} co(s) + \frac{0.1}{s} d(s)$$

$$\text{f. } pv(s) = \frac{0.5}{2s+1} e^{-1.5s} [co(s) + d(s)]$$

$$\text{g. } pv(s) = \frac{-0.5}{s} e^{-1.2s} [co(s) - d(s)]$$

ooOoo

## Bab 2

# KONTROL PID UNTUK PROSES INDUSTRI : BERBAGAI STRUKTUR DAN METODA TUNING PRAKTIS

### Tujuan Bab

Khusus dalam bab ini akan dibahas berbagai hal penting yang terkait dengan PID untuk aplikasi kontrol proses industri. Setelah menyelesaikan bab ini, anda diharapkan dapat:

- Menyebutkan berbagai struktur/algorithm kontrol PID yang umum dijumpai pada modul kontrol komersil beserta metode-metode tuningnya
- Menjelaskan kelebihan dan kelemahan masing-masing struktur kontrol PID tersebut
- Menjelaskan satuan parameter kontrol PID
- Menjelaskan pengaruh besaran parameter kontrol PID terhadap dinamika proses yang dikontrol
- Memilih struktur dan metode tuning PID yang tepat untuk sebuah proses yang dihadapi
- Menentukan besaran parameter kontrol PID yang optimal untuk sebuah proses yang dikontrol
- Menjelaskan pengertian integrator WindUp dan cara antisipasinya
- Menjelaskan struktur Kontrol Smith Predictor
- Melakukan simulasi kontrol PID dengan menggunakan simulink

### Bacaan Prasyarat

- Bab 1. Dinamika dan Model Kontrol Proses Industri

### 2.1 Pendahuluan

Lebih dari setengah abad, PID telah menjadi tulang punggung keberhasilan pengontrolan beragam variabel proses industri. Kontrol PID dapat dijumpai hampir pada setiap industri yang bergerak dalam bidang proses. Menurut sebuah survey dinyatakan bahwa 97 % industri yang bergerak dalam bidang tersebut menggunakan PID dalam pengontrolannya.

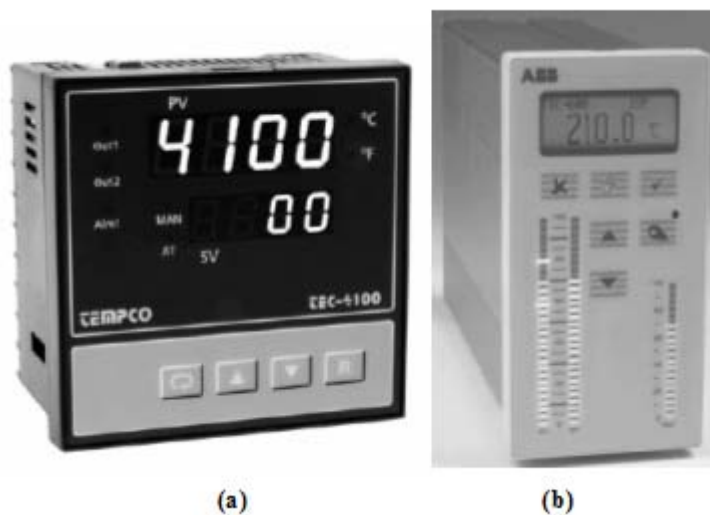
Luasnya penggunaan kontrol PID pada dasarnya dilatarbelakangi beberapa hal, diantaranya:

- Kesederhanaan struktur kontrol: Selain hanya ada tiga parameter utama yang perlu diatur atau dilakukan usaha penalaan (*tuning*), pengaruh perubahan setiap parameter PID terhadap dinamika pengontrolan secara intuitive mudah dipahami oleh operator.
- Kontrol PID memiliki sejarah yang panjang. Dalam hal ini PID telah digunakan jauh sebelum era digital berkembang (yaitu sekitar tahun 1930-an).

- Kontrol PID dalam banyak kasus telah terbukti menghasilkan unjukkerja relative memuaskan, baik digunakan sebagai sistem Regulator (sistem kontrol dengan Setpoint konstan dan beban cenderung berubah-ubah) maupun sebagai sistem Servo (sistem kontrol dengan Setpoint yang berubah dan beban cenderung konstan)

Pada awal penggunaannya, strategi kontrol PID untuk tujuan kontrol proses industri umumnya diimplementasikan dengan menggunakan rangkaian elektronika analog, bahkan banyak diantaranya direalisasikan dengan menggunakan komponen mekanis dan pneumatis murni.

Seiring dengan perkembangan dunia digital (terutama microprocessor), dewasa ini PID dapat dijumpai dalam berbagai bentuk modul komersil, yaitu mulai dari sekedar modul PID untuk pengontrolan satu jenis variabel proses tertentu saja (*special purpose process controller*), sampai modul PID untuk tujuan pengontrolan beragam variabel proses (*general purpose process controller*) atau lebih dikenal dengan nama populer DCS- *Distributed Control System* (lihat gambar 2.1). Bahkan perkembangan terakhir, kontrol PID juga telah banyak ditanamkan pada sistem PLC - *Programmable Logic Controller*.



**Gambar 2.1.** (a) Modul Temperature Controller (*Special Purpose Process Controller*)-Produk Tempco Electric Corporation (b) General Purpose process Controller- Produk ABB

Selain diimplementasikan pada modul-modul kontroler seperti tersebut diatas, algoritma PID juga dapat dijumpai dalam berbagai peralatan yang memerlukan pengontrolan umpan balik kontinyu lainnya (misal peralatan autofokus kamera, servo antenna, pengatur kecepatan otomatis kendaraan, penjejak matahari dan sebagainya).

## 2.2 Berbagai Struktur Kontrol PID (Dasar) dan Metode Tuningnya

Terkait dengan penggunaan modul PID komersil dalam sebuah kontrol proses industri, salah satu hal penting yang perlu diketahui oleh pengguna adalah algoritma atau struktur PID yang tertanam pada modul tersebut – Hal ini dikarenakan algoritma PID dimungkinkan berbeda antar produk satu vendor dengan produk vendor lainnya. Bahkan tidak jarang dalam sebuah modul tertentu, akan dijumpai beragam algoritma atau struktur PID yang dapat dipilih secara bebas.

Ditinjau dari aspek tuning parameter kontrolnya, pengetahuan struktur PID tersebut juga sangat penting, karena struktur PID yang berbeda akan memiliki metode tuning yang relative berbeda pula.

Dengan mengetahui struktur PID pada modul dan bersama-sama pengetahuan model proses yang dikontrol (materi tentang model proses dijelaskan dalam **Bab I**), nilai parameter PID yang optimal pada dasarnya dapat dicari secara mudah berdasarkan metode-metode tuning yang telah dikembangkan sebelumnya oleh sejumlah peneliti dan praktisi bidang kontrol proses, baik secara empiris maupun berdasarkan metode-metode optimasi tertentu.

Berikut ini adalah berbagai struktur atau algoritma PID beserta metode tuning-nya yang dapat dijumpai pada sejumlah modul kontrol PID komersil dan beberapa literatur kontrol.

### 2.2.1 Struktur PID Ideal

Struktur kontrol PID ideal dikenal juga dengan nama struktur PID Tipe A atau struktur PID Paralel atau struktur PID non-interacting. Struktur ini adalah struktur PID yang umum ditemukan pada sebagian besar buku-buku teks kontrol. Persamaan (2.1) dan (2.2) berikut memperlihatkan bentuk umum dari PID ideal.

$$CO(t) = K_P \left[ e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right] \quad (2.1)$$

atau

$$CO(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (2.2)$$

dengan  $CO(t)$  : output kontroler

$e(t)$  : selisih antara SetPoint dengan nilai output proses (dengan menganggap proses yang dikontrol bersifat direct maka  $e = SP - PV$ )

$K_P$  : gain proporsional

$T_I$  : time integral

$T_D$  : time derivative

$K_I$  : gain integral ( $K_P/T_I$ )

$K_D$  : gain derivative ( $K_P T_D$ )

Secara praktis struktur PID pada persamaan (2.1) dan persamaan (2.2) diatas berturut-turut dikenal juga dengan istilah PID ideal bentuk *dependent* dan PID ideal bentuk *independent*. Kedua istilah tersebut mengacu pada depedensi (ketergantungan) setiap suku persamaan terhadap perubahan nilai gain Proporsioanl ( $K_P$ ). Perhatikan dalam bentuk persamaan (2.1), perubahan nilai  $K_P$  tidak saja berpengaruh pada nilai penguatan proporsional, tetapi juga akan mempengaruhi keluaran suku integrator dan suku derivator (sehingga persamaan tersebut dinamakan bentuk *dependent*), sedangkan untuk struktur PID pada persamaan (2.2), perubahan nilai  $K_P$  hanya akan mempengaruhi suku keluaran proporsional saja (keluaran output kontrol dari suku integrator dan derivator tidak terpengaruh -- sehingga dengan demikian persamaan tersebut dinamakan bentuk *independent*).

Dalam kawasan Laplace, persamaan (2.1) dan (2.2) tersebut dapat ditulis:

$$CO(s) = K_P \left[ 1 + \frac{1}{T_I s} + T_D s \right] e(s) \quad (2.3)$$

$$CO(S) = \left[ K_P + \frac{K_I}{s} + K_D s \right] e(S) \quad (2.4)$$

Berhubung kontroler komersil (dalam hal ini *general purpose process controller*) merupakan sebuah modul yang dirancang untuk dapat mengontrol beragam variabel proses (tidak hanya satu jenis saja), satuan-satuan atau terminologi yang digunakan pada kontroler biasanya akan bebas dari satuan fisik variabel yang dikontrol. Berikut ini adalah satuan yang umum digunakan pada kontroler PID komersil:

- Satuan  $K_P$ : %/% (atau tanpa satuan)
- Satuan  $T_I$  : menit atau detik (dalam beberapa modul ditulis menit/siklus atau detik/siklus)
- Satuan  $T_D$  : menit atau detik
- Satuan  $K_I$  : 1/menit atau 1/detik (dalam beberapa modul ditulis siklus/menit atau siklus/detik)
- Satuan  $K_D$  : menit atau detik

Khusus untuk gain Proporsional, beberapa vendor PID komersil seperti Yokogawa dalam produknya menggunakan istilah *Proporsional Band (PB)* yang dirumuskan oleh persamaan (2.5) berikut:

$$PB(\%) = \frac{100\%}{K_P} \quad (2.5)$$

Dalam hal ini, semakin besar nilai  $PB$  maka gain Proporsional yang digunakan akan semakin kecil

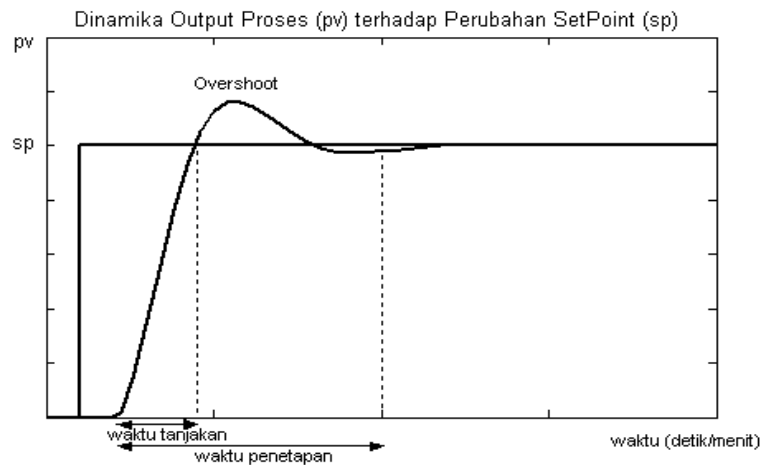
Walaupun secara teoritis nilai-nilai parameter kontrol PID besarnya dapat diatur secara independen, tetapi secara praktis (terutama terkait dengan model proses yang dikontrol), pengaturan atau tuning yang dilakukan terhadap salah satu parameter kontrol, umumnya memerlukan pengaturan ulang pada parameter yang lain. Sebagai contoh, untuk sebuah model proses tertentu, pengubahan nilai  $K_P$  (diperbesar atau diperkecil) umumnya akan memerlukan setting ulang pada besaran parameter kontrol  $K_I$  ( atau  $T_I$ ) dan  $K_D$  (atau  $T_D$ ). Tabel 2.1 dibawah memperlihatkan pengaruh perubahan setiap parameter PID terhadap unjuk kerja pengontrolan proses secara umum.

**Tabel 2.1.** Pengaruh tuning salah satu parameter PID terhadap unjukkerja proses

	Waktu Tanjakan	Overshoot	Waktu Penetapan	Kestabilan
Pembesaran $K_P$	Berkurang	Bertambah	Sedikit bertambah	Menurun
Pembesaran $K_I$ (Pengecilan $T_I$ )	Sedikit berkurang	Bertambah	Bertambah	Menurun
Pembesaran $K_D$ (Pembesaran $T_D$ )	Sedikit berkurang	Berkurang	Berkurang	Meningkat

Dalam bentuknya yang ideal, parameter unjukkerja yang nampak pada tabel 2.1 diatas dapat dijelaskan sebagai berikut (lihat juga ilustrasinya pada gambar 2.2):

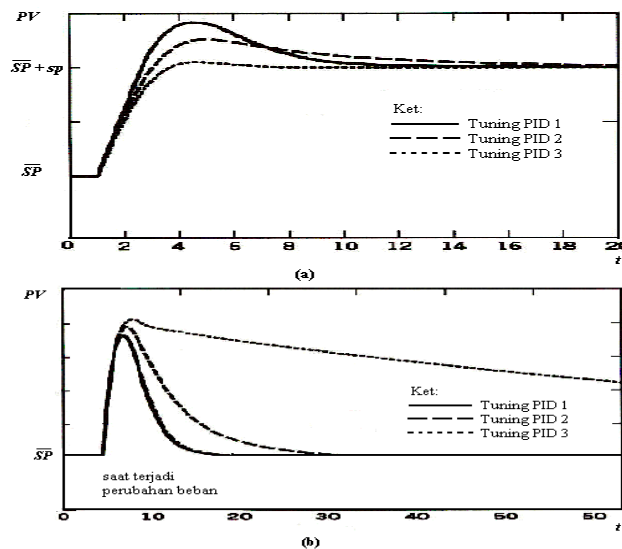
- **Waktu tanjakan:** Waktu yang diperlukan respon (deviasi output variable proses) untuk naik dari 0 sampai 100% harga akhirnya.
- **Overshoot:** Lonjakan maksimum yang dialami oleh respon proses.
- **Waktu penetapan:** Waktu yang diperlukan respon untuk mencapai dan menetap disekitar 95%-98% dari harga akhirnya.



**Gambar 2.2.** Respon proses sebagai akibat perubahan SetPoint

Terkait dengan metode tuning parameter kontrol PID, salah satu hal yang perlu diperhatikan adalah problem yang dihadapi pada pengontrolan proses tersebut. Dalam hal ini, perlu diketahui terlebih dulu apakah proses yang dikontrol selama operasinya sering mengalami perubahan titik operasi (problem Servo) atau apakah proses tersebut sering mengalami perubahan beban (problem Regulator).

Hal diatas dikarenakan hasil tuning parameter PID yang optimal untuk problem Servo belum tentu menghasilkan unjuk kerja yang memuaskan pada problem Regulator (hal yang sama berlaku juga sebaliknya -lihat gambar 2.3).



**Gambar 2.3.** (a) Problem servo: Respon proses dalam menanggapi perubahan SetPoint (b) Problem Regulator: Respon proses dalam menanggapi perubahan beban (gangguan)



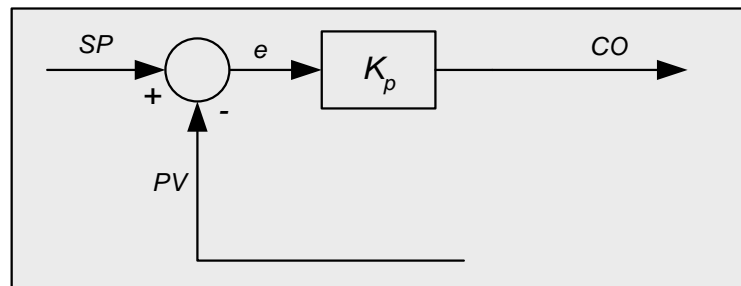
Tergantung nilai setting ke-tiga parameter PID, dibawah ini empat jenis kontrol yang dapat digunakan dalam pengontrolan variabel proses:

- Kontrol Proporsional (Kontrol P)
- Kontrol Proporsional Integral (Kontrol PI)
- Kontrol Proporsional Derivative (kontrol PD)
- Kontrol Proporsional Integral Derivative (Kontrol PID)

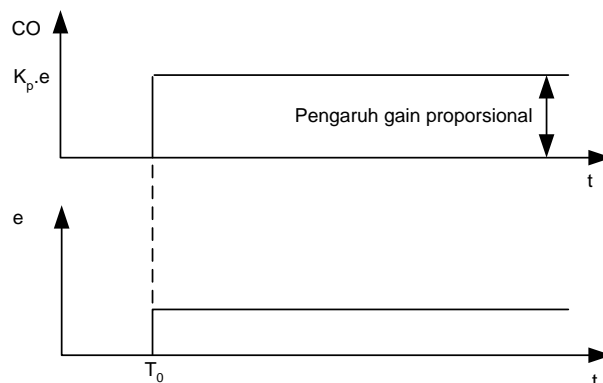
#### • Kontrol Proporsional (P)

Berdasarkan Persamaan (2.1), kontrol Proporsional secara praktis dapat diperoleh dengan cara men-*setting* nilai  $T_I$  dan  $T_D$  berturut – turut sama dengan nilai tak hingga dan nol (atau  $K_I = K_D = 0$  -- lihat persamaan (2.2)). Sehingga dengan demikian, hubungan antara output kontrol-  $CO$  dengan error penggerak pada aksi kontrol Proporsional dapat ditulis:

$$CO(t) = K_p e(t) \quad (2.6)$$



**Gambar 2.4.** Blok kontrol Proporsional

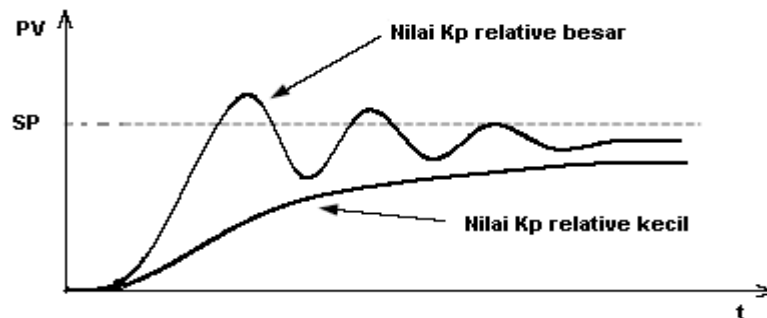


**Gambar 2.5.** Relasi input-output kontrol Proporsional

Dari persamaan (2.6), nampak bahwa besar output kontrol ( $CO$ ) akan selalu sebanding dengan besar sinyal error penggerak. Gambar 2.4 dan 2.5 berturut-turut memperlihatkan blok diagram kontrol Proporsional dan relasi input-output -nya.

Khusus untuk pengontrolan proses FOPDT, aksi kontrol Proporsional dalam menanggapi perubahan Setpoint dan/atau gangguan secara praktis akan selalu menunjukkan offset (error steady) pada output proses yang dikontrolnya. Besar-kecilnya error *steady* sangat tergantung pada nilai penguatan (gain) yang dipilih: Semakin besar  $K_P$  yang digunakan, maka error *steady* akan semakin mengecil. Pembesaran nilai gain tersebut selain akan memperkecil error *steady*, juga akan mempercepat respon output proses (memperkecil

waktu penetapan proses). Tetapi walaupun demikian, pembesaran nilai  $K_P$  secara praktis tidak dapat dilakukan secara terus menerus karena pada penguatan tertentu, output proses akan menjadi relative kurang bahkan tidak stabil (lihat gambar 2.6).



**Gambar 2.6.** Pengaruh besar gain proporsional terhadap dinamika output proses FOPDT

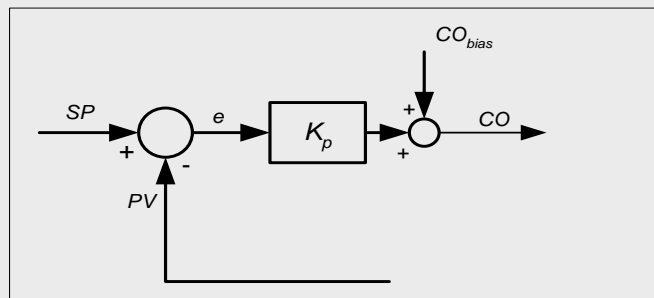
#### Catatan Penting tentang offset atau error steady

Dalam sistem kontrol umpan balik (*closed loop*), error steady akan selalu muncul jika model proses yang dikontrol atau kontroler-nya itu sendiri tidak memiliki komponen yang memiliki sifat integrator (misal dalam hal ini pengontrolan model proses FOPDT dengan menggunakan kontrol Proporsional).

Secara praktis phenomena offset tersebut dapat dihilangkan dengan cara menambahkan sinyal kontrol bias pada output kontrol Proporsionalnya.

Persamaan 2.7 dan gambar 2.7 berikut berturut-turut memperlihatkan persamaan kontrol Proporsional dan diagram bloknya dengan mencantumkan sinyal kontrol bias secara eksplisit.

$$CO(t) = K_p e(t) + CO_{bias} \quad (2.7)$$



**Gambar 2.7.** Diagram blok kontrol Proporsional dengan bias

Untuk nilai SetPoint dan beban proses tertentu, besar sinyal bias dapat diatur atau di-setting secara manual oleh operator sedemikian sehingga nilai output proses ( $PV$ ) akan sama dengan nilai SetPoint. Jika satu saat SetPoint atau beban yang dikontrol berubah, maka operator harus men-setting ulang kembali besar sinyal kontrol bias tersebut.

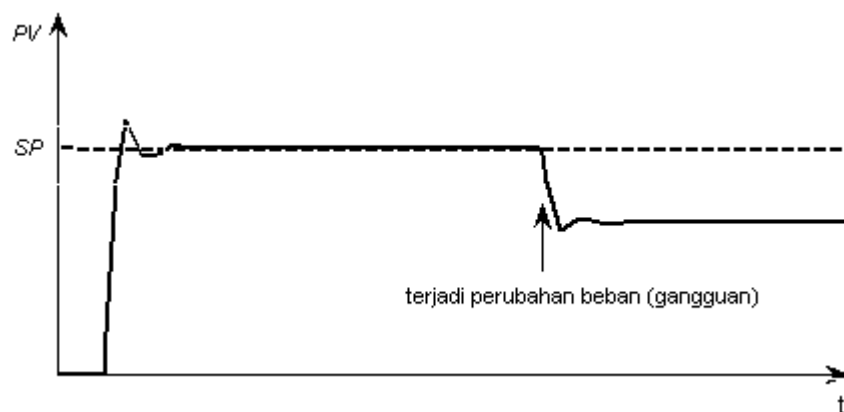
Dalam kaitannya dengan pengontrolan proses FOPDT, besar penguatan  $K_P$  yang harus dipilih secara umum dapat merujuk pada tiga parameter proses FOPDT itu sendiri:

- Kesensitifan proses yang dikontrol (secara teoritis ditunjukkan oleh besar nilai gain statis proses:  $K$ ): Semakin sensitive sebuah proses (nilai gain statis proses

semakin besar) maka nilai  $K_P$  yang di-*setting* harus semakin kecil (berbanding terbalik)

- Waktu ketidakpastian respon proses (ditunjukkan oleh nilai delay transportasi proses  $-L$ ): semakin lama waktu ketidakpastian (delay transportasi), maka nilai  $K_P$  yang di-*setting* harus semakin kecil (berbanding terbalik).
- Kecepatan respon proses (ditunjukkan oleh nilai Konstanta waktu proses  $-T$ ): semakin cepat sebuah proses (atau semakin kecil  $T$ ), maka nilai  $K_P$  yang di-*setting* harus semakin kecil (berbanding lurus)

Berbeda dengan respon model proses FOPDT dalam menanggapi aksi kontrol Proporsional. Berdasarkan sifat prosesnya (yaitu memiliki integrator dalam), penerapan aksi kontrol Proporsional pada model proses IPDT secara praktis akan menghasilkan offset atau error steady hanya jika proses yang dikontrol mengalami gangguan atau perubahan beban saja (lihat ilustrasinya pada gambar 2.8)



**Gambar 2.8.** Pengaruh perubahan setpoint dan beban (gangguan) pada sistem kontrol proporsional untuk proses IPDT

Terkait dengan gain Proporsional yang harus di-*setting*, penentuan besar gain proporsional untuk model IPDT ini memiliki aturan yang relative sama dengan kasus pengontrolan model FOPDT:

- Semakin sensitif model IPDT yang akan dikontrol (ditunjukkan oleh besar nilai gain integrative proses  $-K^*$ ) maka gain Proporsional yang di-*setting* harus semakin kecil (berbanding terbalik)
- Semakin besar waktu ketidakpastian model IPDT (ditunjukkan oleh besar waktu keterlambatan transportasi  $-L$ ) maka gain Proporsional yang di-*setting* harus semakin kecil (berbanding terbalik)

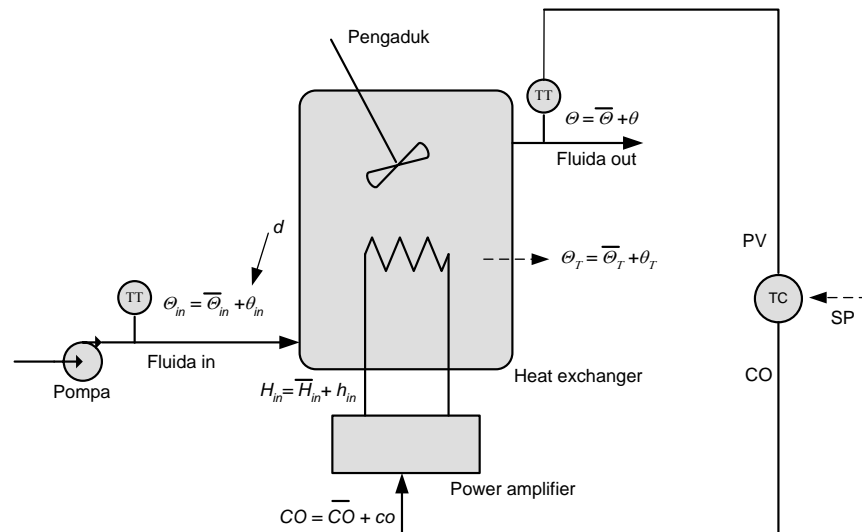
Tabel 2.2 dan tabel 2.3 berikut berturut-turut memperlihatkan metode tuning yang dapat digunakan untuk mendapatkan besar gain proporsional optimal pada pengontrolan proses FOPDT dan IPDT.

**Tabel 2.2.** Metode tuning kontrol Proporsional untuk model proses FOPDT

Metode	$K_P$	Keterangan
Ziegler and Nichols	$\frac{T}{L.K}$	$0.1 < \frac{L}{T} < 1$

**Tabel 2.3.** Metode tuning kontrol Proporsional untuk model proses IPDT

Metode	$K_p$	Keterangan
Ziegler and Nichols	$\frac{0.78}{L.K^*}$	Metode ultimate gain



**Gambar 2.9.** Pengontrolan temperature outlet pada sistem Heat Exchanger

Sebagai bahan studi kasus dalam melihat unjuk kerja aksi kontrol proporsional pada pengontrolan sebuah proses, disini akan diambil contoh pengontrolan temperature outlet sistem *heat exchanger* (model FOPDT) seperti nampak pada gambar 2.9.

Berdasarkan hasil penyelesaian **contoh soal 1.8 dalam Bab 1**, untuk besar parameter proses berikut:

- $V$  (Volume tangki) =  $0.2 \text{ m}^3$
- $\rho$  (densitas fluida) =  $1000 \text{ kg/m}^3$
- $c$  (Kalor jenis fluida) =  $4000 \text{ J/(Kg.}^\circ\text{C)}$
- $K_{ps}$  (konstanta kesebandingan antara perubahan panas dengan perubahan sinyal kontrol) =  $800 \text{ W/\%}$  ( $800.60 \text{ J/menit/\%} = 48000 \text{ J/menit/\%}$ )
- $K_L = 50 \text{ (Kg)}$
- $w$  (laju aliran) =  $20 \text{ kg/menit}$

Sistem *heat exchanger* tersebut memiliki persamaan dinamika :

$$pv(s) = \frac{0.6e^{-2.5s}}{10s + 1} co(s) + \frac{e^{-2.5s}}{10s + 1} d(s) \quad (2.8)$$

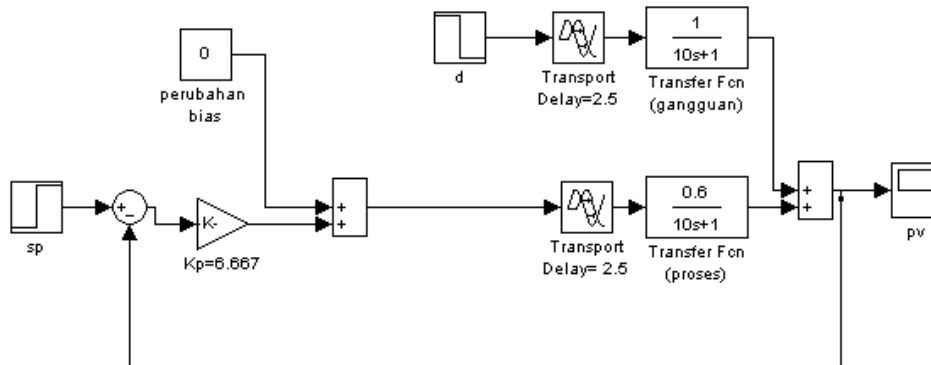
dengan  $pv$  : deviasi temperature outlet disekitar nilai steady nominalnya ( $^\circ\text{C}$ )

$co$  : perubahan output kontroler dari nilai steady awal (%)

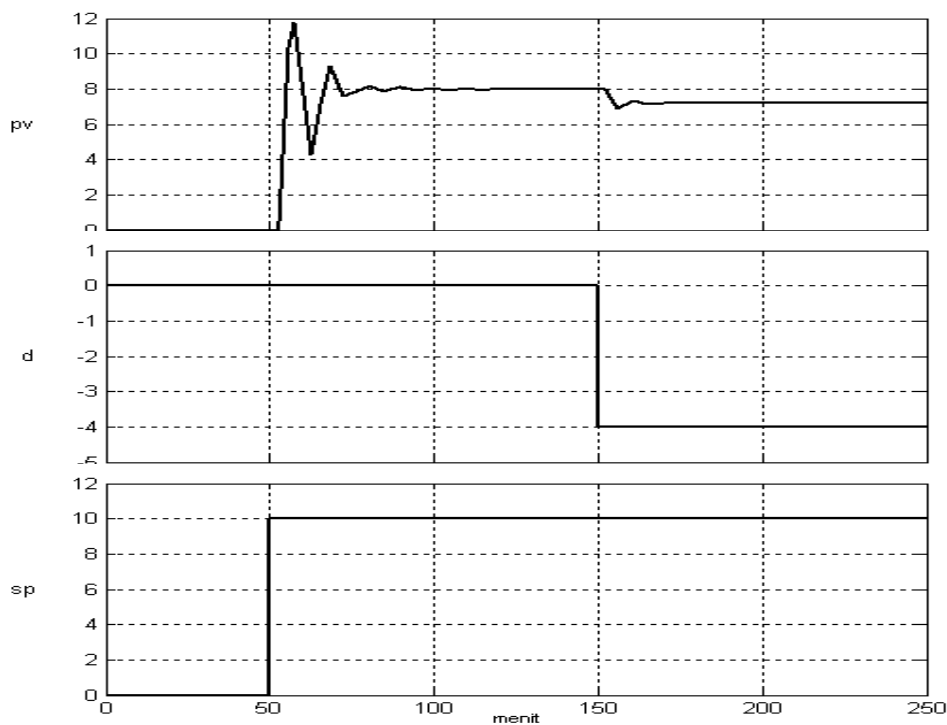
$d$  : deviasi temperature fluida inlet dari nilai steady awal -- berlaku sebagai gangguan ( $^\circ\text{C}$ ).

Untuk melihat dinamika perubahan output proses- $p_v$  terhadap perubahan Setpoint- $s_p$  atau input gangguan –  $d$  disekitar *steady state*-nya, maka cara termudah adalah melakukannya lewat simulasi komputer dengan simulink.

Gambar 2.10 dan 2.11 berikut berturut-turut memperlihatkan diagram blok kontrol proporsional untuk sistem *Heat Exchanger* dalam Simulink beserta salah satu hasil simulasinya (dalam blok diagram Simulink ini, gain proporsional di-*setting* sebesar 6.667 --sesuai dengan hasil perhitungan metode Ziegler-Nichols- lihat tabel 2.2).



**Gambar 2.10.** Diagram blok simulink pengendalian sistem heat exchanger dengan kontrol Proporsional (plus bias manual)



**Gambar 2.11.** Salah satu hasil simulasi simulink gambar 2.10

Berdasarkan hasil simulasi gambar 2.11, nampak bahwa:

- (1). Perubahan output proses ( $p_v$ ) yang disebabkan oleh perubahan SetPoint (problem servo) akan menunjukkan dinamika transien yang berbeda dengan

perubahan output proses yang disebabkan oleh input gangguan (problem regulator): pada gambar tersebut terlihat pada keadaan transiennya, output  $PV$  sebagai akibat perubahan SetPoint menunjukkan *overshoot* (*undershoot*) relative besar dibandingkan untuk kasus input gangguan.

- (2). Sesuai dengan teori, untuk setiap perubahan setpoint yang diberikan (atau terjadinya gangguan), output proses akan selalu menunjukkan selisih atau offset dari perubahan setpoint tersebut. Pada gambar nampak bahwa output proses hanya mengalami perubahan sebesar 8 (°C), untuk perubahan SetPoint sebesar 10 (°C). Selisih yang terjadi secara praktis hanya dapat diminimalkan atau dihilangkan dengan cara mengatur ulang besar sinyal bias manual pada output kontrolernya (dalam simulasi gambar 2.10, hal tersebut tidak dilakukan)

#### • Kontrol Proporsional Integral (PI)

Dalam aplikasinya di lapangan, Kontrol Proporsional Integral biasanya digunakan untuk pengontrolan proses yang memiliki dinamika relative cepat ( yaitu seperti aliran, tekanan dan level). Berdasarkan sebuah survey dinyatakan bahwa hampir 80% kontroler PID yang terinstal di industri menggunakan kontrol PI dalam operasinya.

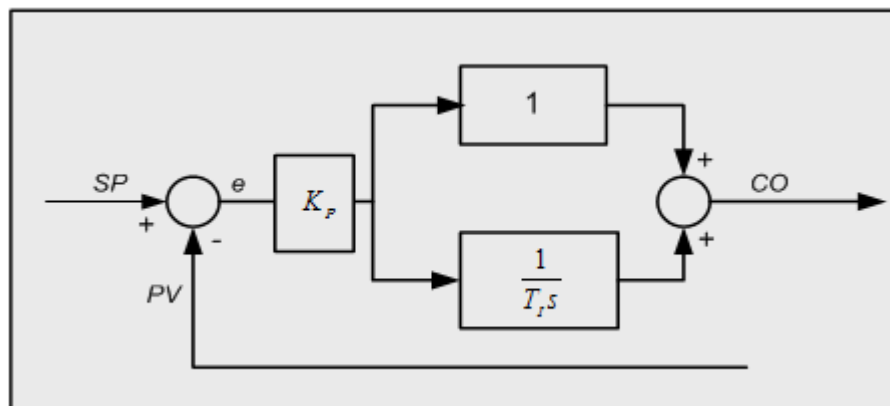
Untuk sebuah modul kontrol PID, Kontrol PI dapat diperoleh dengan cara men-*setting* nilai gain (waktu) derivative sama dengan nol, atau secara matematis dapat ditulis:

$$CO(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right] \quad (2.9)$$

Dalam kawasan Laplace, persamaan 2.9 tersebut dapat direpresentasikan:

$$CO(s) = K_p \left[ 1 + \frac{1}{T_i s} \right] e(s) \quad (2.10)$$

Diagram blok untuk perumusan diatas diperlihatkan pada gambar (2.12) berikut:

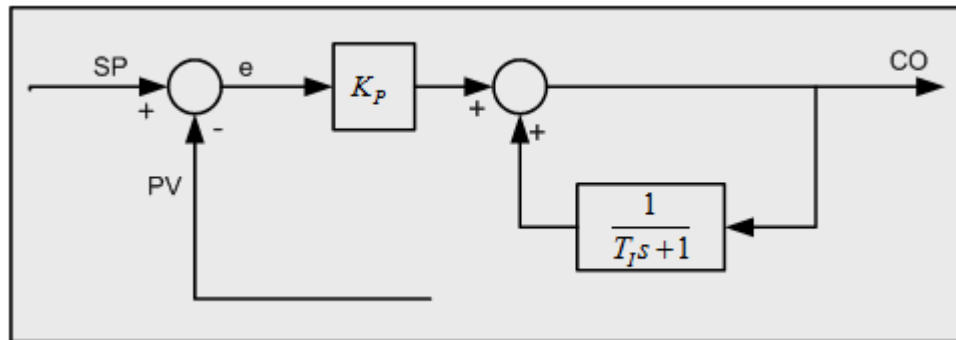


**Gambar 2.12.** Diagram blok kontrol Proporsional Integral

Penambahan blok integrator di dalam kontroler pada dasarnya dimaksudkan untuk menggantikan sinyal bias manual (atau manual reset) yang berfungsi menghilangkan error *steady* (lihat kembali gambar 2.7): Selama masih terdapat selisih atau error antara

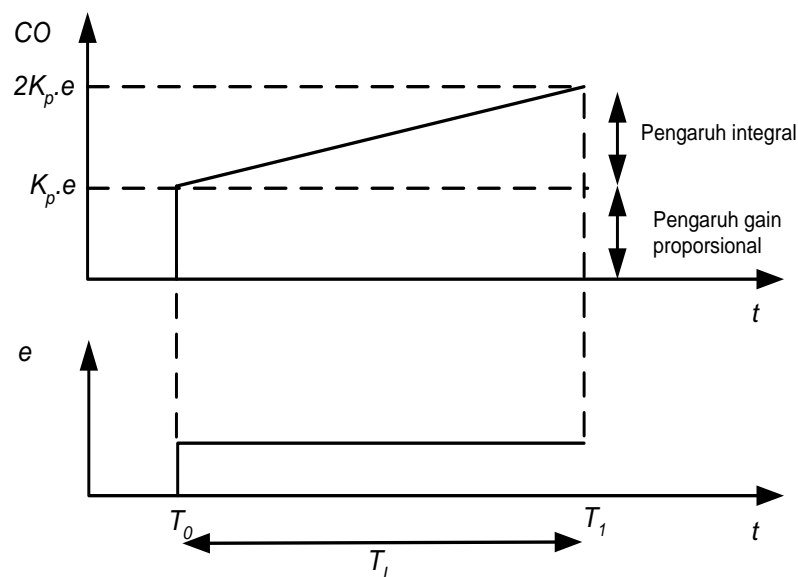
Setpoint dengan variabel proses maka output kontroler PI akan selalu membesar atau mengecil.

Selain gambar 2.12. kontrol PI dapat juga direalisasikan dalam bentuk diagram blok seperti nampak pada gambar 2.13 dibawah.



Gambar 2.13. Diagram blok alternative kontrol PI

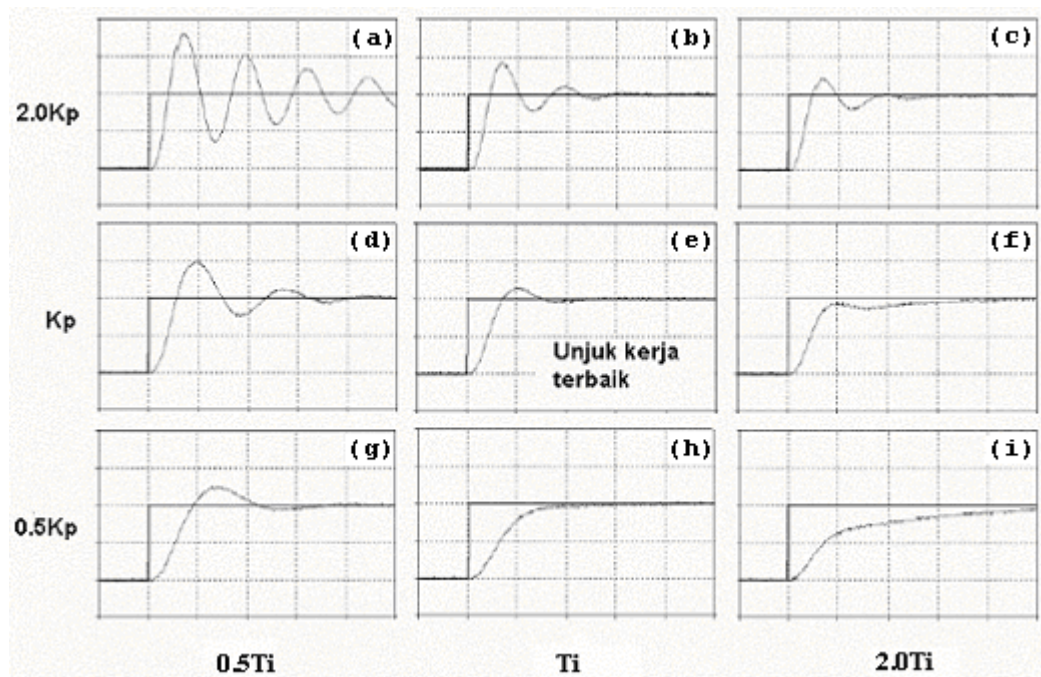
Salah satu kelebihan struktur PI alternatif pada gambar 2.14 tersebut adalah dapat disisipkannya model saturasi dalam kontroler yang berfungsi sebagai antiwindup (lihat subbab 2.4 AntiWindup)



Gambar 2.14. Aksi kontrol Proporsional Integral (PI)

Untuk nilai  $K_p$  dan  $T_i$  yang dipilih, kontribusi suku integral akan sama nilainya dengan pengaruh kontrol proporsional setelah selang waktu  $T_i$  (lihat gambar 2.14).

Seperti halnya dengan gain Proporsional, pemilihan parameter waktu (atau gain) integral pada kontrol PI secara praktis akan sangat mempengaruhi dinamika keluaran variabel output proses yang sedang dikontrol. Jika kedua parameter kontrol ( $K_p$  dan  $T_i$ ) ini dipilih secara tepat, maka respon yang didapat akan optimal (sesuai harapan). Pembesaran atau pengecilan nilai  $K_p$  dan  $T_i$  dari nilai optimalnya dapat menyebabkan respon menjadi terlalu lambat atau terlalu berosilasi. Gambar 2.15 dibawah memperlihatkan pengaruh pembesaran (pengecilan)  $K_p$  dan  $T_i$  dari nilai optimalnya terhadap dinamika output.



**Gambar 2.15.** Pengaruh pembesaran (pengecilan)  $K_p$  dan  $T_i$  terhadap respon output proses.

Berdasarkan ilustrasi gambar 2.15 tersebut, dapat dilihat bahwa untuk nilai  $K_p$  yang tetap, pengecilan nilai  $T_i$  (atau pembesaran  $K_i$ ) dari nilai optimalnya akan menyebabkan respon proses semakin berosilasi dengan perioda osilasi yang relative lama, sedangkan jika nilai  $T_i$  tersebut diperbesar (atau nilai  $K_i$  diperkecil) dari nilai optimalnya, maka hal tersebut akan menyebabkan respon proses akan semakin lambat (*sluggish*) dalam menanggapi perubahan SetPoint (atau merespon gangguan).

Hal yang relative sama terjadi pada kasus pembesaran gain proporsional dari nilai optimalnya (hanya saja perioda osilasi yang dihasilkan relative lebih cepat). Berdasarkan gambar 2.16 diatas, dapat dilihat juga respon output proses jika kedua parameter kontroler tersebut diubah secara simultan.

Sekarang, pertanyaannya adalah Bagaimana parameter kontroler yang optimal diperoleh?. Seperti pada kasus kontrol Proporsional, nilai parameter PI yang optimal itu sendiri pada dasarnya sangat tergantung dari model dan nilai parameter proses yang dikontrol. Tabel 2.4 dan 2.5 berikut berturut-turut memperlihatkan sejumlah metode yang dapat digunakan sebagai tuning awal nilai paramater kontrol PI untuk model proses FOPDT dan model proses IPDT.

**Tabel 2.4.** Beberapa metode tuning kontrol Proporsional Integral untuk proses FOPDT

Metode	$K_p$	$T_i$	Keterangan
Chien – regulator 1	$\frac{0.6T}{K.L}$	$4L$	0% overshoot - $0.11 < \frac{L}{T} < 1.0$
Chien – regulator 2	$\frac{0.7T}{K.L}$	$2.33 \frac{L}{K}$	20% overshoot - $0.11 < \frac{L}{T} < 1.0$
Astrom and Hagglund - regulator	$\frac{0.7T}{K.L}$	$2.3L$	20% overshoot
Chien – servo 1	$\frac{0.35T}{K.L}$	$1.17T$	0% overshoot -



			$0.11 < \frac{L}{T} < 1.0$
Chien – servo 2	$\frac{0.6T}{K.L}$	$T$	20% overshoot- $0.11 < \frac{L}{T} < 1.0$
Ziegler and Nichols process reaction	$\frac{0.9T}{K.L}$	$3.33L$	Quarter decay ratio $\frac{L}{T} \leq 1$
Two constraints criterion - Murrill	$\frac{0.928}{K} \left( \frac{T}{L} \right)^{0.946}$	$\frac{T}{1.078} \left( \frac{L}{T} \right)^{0.583}$	Quarter decay ratio (servo mode). $0.1 \leq \frac{L}{T} \leq 1.0$
St. Clair	$\frac{0.333T}{K.L}$	$T$	$\frac{L}{T} \leq 3.0$
Minimum IAE - Murrill Regulator	$\frac{0.984}{K} \left( \frac{T}{L} \right)^{0.986}$	$\frac{T}{0.608} \left( \frac{L}{T} \right)^{0.707}$	$0.1 \leq \frac{L}{T} \leq 1.0$
Minimum ISE – Zhuang and Atherton	$\frac{1.279}{K} \left( \frac{T}{L} \right)^{0.945}$	$\frac{T}{0.535} \left( \frac{L}{T} \right)^{0.586}$	$0.1 \leq \frac{L}{T} \leq 1.0$
	$\frac{1.346}{K} \left( \frac{T}{L} \right)^{0.675}$	$\frac{T}{0.552} \left( \frac{L}{T} \right)^{0.438}$	$1.1 \leq \frac{L}{T} \leq 2.0$
Minimum ITAE - Murrill	$\frac{0.859}{K} \left( \frac{T}{L} \right)^{0.977}$	$\frac{T}{0.674} \left( \frac{L}{T} \right)^{0.680}$	$0.1 \leq \frac{L}{T} \leq 1.0$

**Tabel 2.5.** Beberapa metode tuning kontrol Proporsional Integral untuk proses IPDT

Metode	$K_P$	$TI$	Keterangan
Ziegler and Nichols	$\frac{0.9}{K^* . L}$	$3.3L$	Quarter decay ratio
Tyreus Luyben	$\frac{0.487}{K^* . L}$	$8.75L$	Konstanta closed loop $\sqrt{10}L$
Astrom and Hagglund - regulator	$\frac{0.63}{K^* . L}$	$3.2L$	Ekivalen dengan ultimated cycle Ziegler-Nichols
Shinkey	$\frac{0.9259}{K^* . L}$	$4L$	Minimum IAE regulator
Poulin	$\frac{0.5264}{K^* . L}$	$4.58L$	Minimum ITAE regulator
Poulin	$\frac{0.9259}{K^* . L}$	$3.88L$	Minimum ITAE Servo

Hal penting yang perlu diperhatikan dari kedua tabel tersebut adalah pada bagian kolom "keterangan": Walaupun disana misalnya tertulis respon output 0% *overshoot*, 20% *overshoot* dan sebagainya, tetapi respon proses sebenarnya mungkin saja sedikit berbeda dengan apa yang tercantum dalam keterangan tabel tersebut (hal ini disebabkan respon proses sesungguhnya sangat tergantung pada nilai eksak dari masing-masing parameter proses yang dikontrol).

Terkait dengan metode tuning yang akan digunakan atau dipilih dari tabel, terlebih dulu perlu diketahui problem yang dihadapi oleh proses : Jika selama operasi berlangsung,

beban yang dikontrol sering berubah (atau proses sering mengalami gangguan) maka untuk tuning awal sebaiknya dipilih metode tuning yang memang dikhususkan untuk problem regulator, dan begitu juga sebaliknya, jika selama operasi berlangsung, titik kerja sering mengalami perubahan maka untuk tuning awal sebaiknya digunakan metode tuning untuk problem servo.

Tetapi walaupun demikian, untuk sebuah proses tertentu, nilai parameter kontrol ( $K_P$  dan  $T_I$ ) yang didapat untuk setiap metode tuning pada dasarnya tidak jauh berbeda, hal tersebut disebabkan perumusan setiap metode hanya berbeda dalam faktor pembobotannya saja.

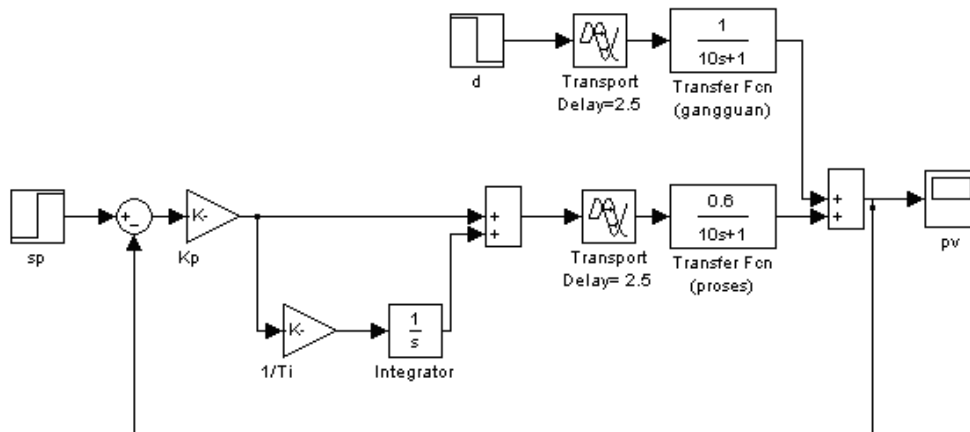
Sebagai gambaran, tabel 2.6 dibawah memperlihatkan hasil perhitungan parameter kontrol PI untuk tiap metode tuning pada kasus pengontrolan *Heat Exchanger* persamaan 2.8 sebelumnya.

**Tabel 2.6.** Parameter kontrol PI hasil perhitungan setiap metode pada kasus sistem kontrol Heat Exchanger

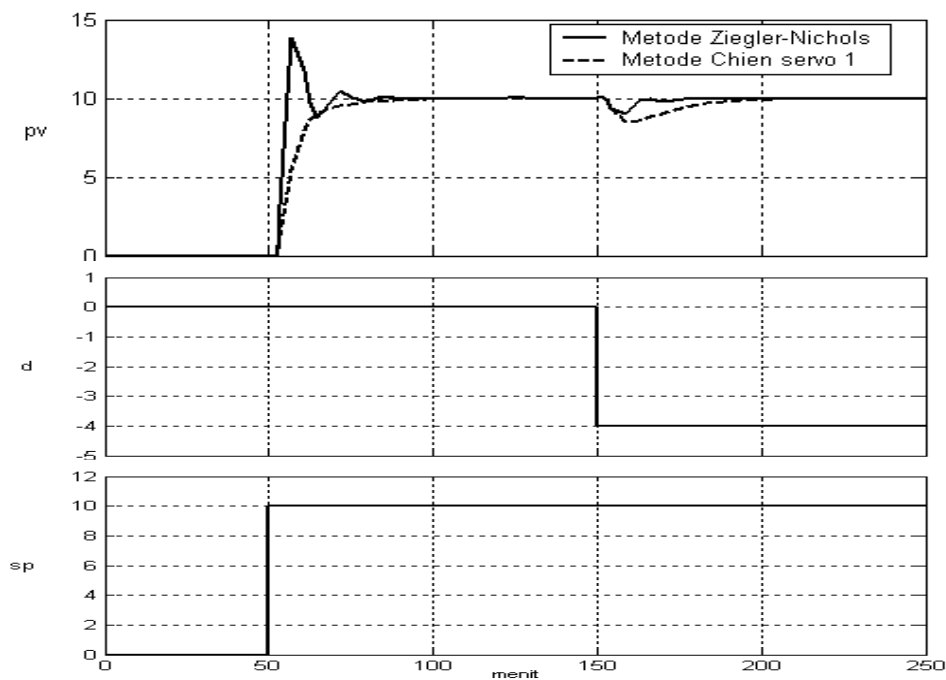
Metode	Kp	Ti	Keterangan
Chien – regulator 1	4	10	0% overshoot - $0.11 < \frac{L}{T} < 1.0$
Chien – regulator 2	4.667	9.7	20% overshoot - $0.11 < \frac{L}{T} < 1.0$
Astrom and Hagglund - regulator	4.667	5.7	20% overshoot
Chien – servo 1	2.3	11.7	0% overshoot - $0.11 < \frac{L}{T} < 1.0$
Chien – servo 2	4	10	20% overshoot- $0.11 < \frac{L}{T} < 1.0$
Ziegler and Nichols process reaction	6	8.325	Quarter decay ratio $\frac{L}{T} \leq 1$
Two constraints criterion - Murrill	5.74	4.13	Quarter decay ratio (servo mode). $0.1 \leq \frac{L}{T} \leq 1.0$
St. Clair	2.22	10	$\frac{L}{T} \leq 3.0$
Minimum IAE - Murrill Regulator	6.43	6.17	$0.1 \leq \frac{L}{T} \leq 1.0$
Minimum ISE – Zhuang and Atherton	7.9	8.29	$0.1 \leq \frac{L}{T} \leq 1.0$
	5.7	9.87	$1.1 \leq \frac{L}{T} \leq 2.0$
Minimum ITAE - Murrill	5.54	5.78	$0.1 \leq \frac{L}{T} \leq 1.0$

Cara praktis untuk melihat unjuk kerja hasil perhitungan tiap metode tuning yang tersaji pada tabel diatas adalah dengan mengujinya secara langsung lewat *tool* Simulink.

Gambar 2.16 dan 2.17 berikut berturut-turut memperlihatkan blok simulink kontrol PI untuk sistem *Heat Exchanger* tersebut dan salah satu hasil simulasinya (Sebagai bahan perbandingan, dalam simulasi ini disajikan respon proses untuk dua metode yang berbeda: metode Chien servo 1 dan metode Ziegler-Nichols --lihat tabel 2.6).



**Gambar 2.16.** Diagram blok simulink pengontrolan sistem Heat Exchanger dengan kontrol PI



**Gambar 2.17.** Salah satu hasil simulasi simulink gambar 2.16

Berdasarkan hasil simulasi, nampak bahwa

1. Penggunaan integrator didalam kontroler secara praktis akan menghilangkan error *steady* antara Setpoint dengan output proses.
2. Respon transien yang terjadi sebagai akibat perubahan Setpoint atau input gangguan sangat ditentukan oleh metode tuning yang digunakan. Relative terhadap metode Chien servo 1, dalam menanggapi perubahan setpoint nampak tuning metode Ziegler-Nichols menghasilkan *overshoot* yang terlalu berlebih, tetapi disisi lain ketika terjadi gangguan, metode tuning Ziegler - Nichols ini menghasilkan waktu pemulihan yang relative cepat (Berdasarkan kenyataan tersebut dapat disimpulkan bahwa metode Ziegler-Nichols cocok digunakan untuk kasus kontrol problem regulator).

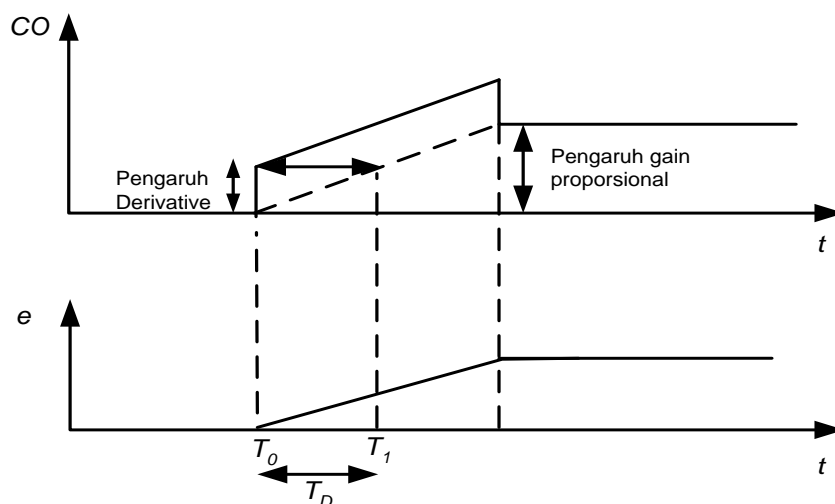
### • Mode Kontrol Proporsional Derivative (PD)

Dalam modul kontrol PID, Kontrol PD diperoleh dengan cara men-setting penguatan integral pada struktur PID sama dengan nol:

$$CO(t) = K_p \left[ e(t) + T_D \frac{de(t)}{dt} \right] \quad (2.11)$$

Sifat utama dari aksi kontrol PD adalah mendahului sinyal masukan (error): Dalam hal ini waktu derivative ( $T_D$ ) adalah waktu yang diperlukan sehingga keluaran Kontrol Proporsional besarnya sama dengan keluaran dari Kontrol Derivative (lihat gambar 2.18).

Secara umum dapat dikatakan bahwa fungsi dari kontrol derivative adalah memperbaiki stabilitas dan mempercepat respon output proses. Tetapi walaupun demikian penggunaan kontrol PD untuk pengontrolan proses (dalam hal ini model FOPDT) secara praktis jarang digunakan karena penggunaan mode kontrol PD ini akan selalu menghasilkan error *steady* (kecuali untuk proses-proses yang secara matematis memiliki integrator internal, yaitu seperti model IPDT).



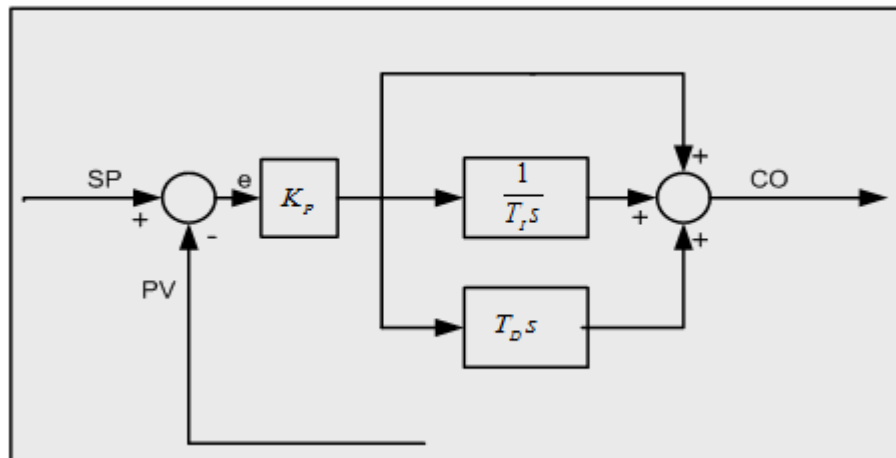
Gambar 2.18. Aksi kontrol Proporsional Derivative

### • Mode Kontrol Proporsional Integral Derivative (PID)

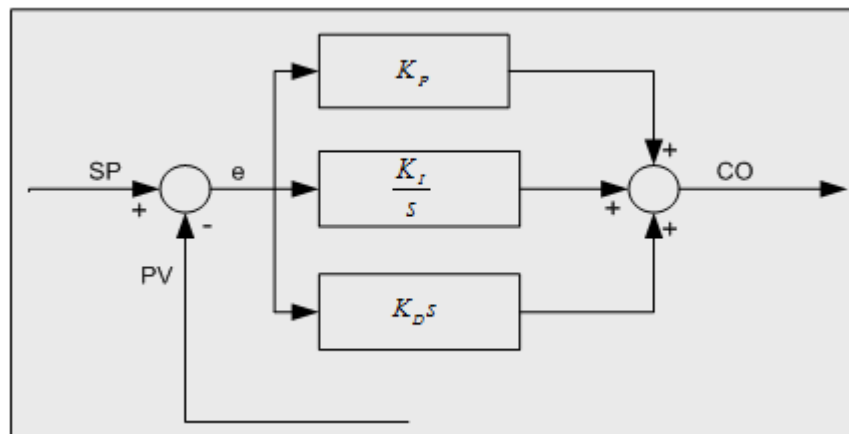
Aksi kontrol PID pada dasarnya bertujuan untuk menggabungkan kelebihan komponen-komponen dasar kontrol PID:

- ✓ Kontrol Proporsional : berfungsi untuk mempercepat respon
- ✓ Kontrol Integral : berfungsi untuk menghilangkan error steady
- ✓ Kontrol Derivative : berfungsi untuk memperbaiki sekaligus mempercepat respon transien.

Gambar 2.19 dan 2.20 berikut berturut-turut memperlihatkan blok diagram kontrol PID ideal bentuk *dependent* dan bentuk *independent* (lihat algoritmanya dalam persamaan 2.3 dan 2.4):

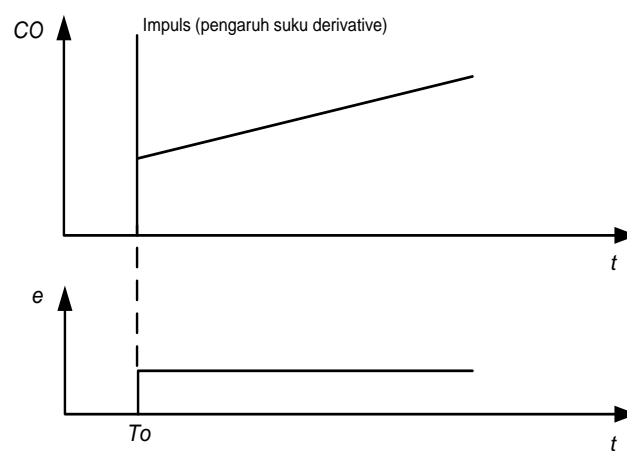


**Gambar 2.19.** Diagram blok kontrol PID ideal bentuk dependent



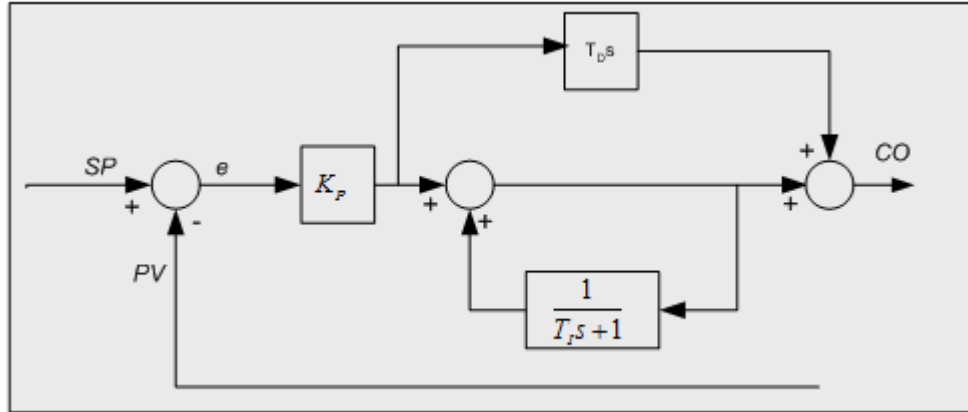
**Gambar 2.20.** Diagram blok kontrol PID ideal bentuk independent

Seperti nampak pada gambar 2.19 dan 2.20 diatas, semua komponen pada kontroler PID jenis ideal ini bekerja pada masukan error penggerak. Hubungan antara masukan error penggerak dengan output kontroler PID tersebut dapat dilihat pada gambar 2.21.



**Gambar 2.21.** Aksi kontrol PID

Selain gambar 2.19 atau gambar 2.20, kontrol PID ideal juga dapat direalisasikan dengan mengacu pada gambar 2.22 berikut:



**Gambar 2.22.** Diagram blok alternatif kontrol PID

Terkait dengan nilai parameter kontroler yang harus dipilih, tabel 2.7 dan 2.8 dibawah berturut-turut memperlihatkan beberapa metode tuning empiris yang dapat digunakan pada pengontrolan model proses FOPDT dan model proses IPDT.

**Tabel 2.7.** Beberapa metode tuning kontrol PID ideal untuk proses FOPDT

Metode	$K_p$	$T_I$	$T_D$	Keterangan
Ziegler -Nichols	$\frac{1.2T}{KL}$	$2.L$	$0.5L$	Quarter decay ratio
Chien Regulator	$\frac{0.95T}{KL}$	$2.38L$	$0.42L$	0% overshoot $0.11 < \frac{L}{T} < 1$
	$\frac{1.2T}{KL}$	$2L$	$0.42L$	20% overshoot $0.11 < \frac{L}{T} < 1$
Chien Servo	$\frac{0.6T}{KL}$	$T$	$0.5L$	0% overshoot $0.11 < \frac{L}{T} < 1$
	$\frac{0.95T}{KL}$	$1.36L$	$0.47L$	20% overshoot $0.11 < \frac{L}{T} < 1$
Astrom-Haglung	$\frac{0.94T}{KL}$	$2L$	$0.5L$	Ekuivalen Ultimated cycle ZN
Minimum IAE Muril regulator	$\frac{1.435}{K} \left[ \frac{T}{L} \right]^{0.921}$	$\frac{T}{0.878} \left[ \frac{L}{T} \right]^{0.746}$	$0.482T \left[ \frac{L}{T} \right]^{1.137}$	$0.11 < \frac{L}{T} < 1$
Modified IAE Cheng dan Hung	$\frac{3}{K} \left[ \frac{T}{L} \right]^{0.921}$	$\frac{T}{0.878} \left[ \frac{L}{T} \right]^{0.749}$	$0.482T \left[ \frac{L}{T} \right]^{1.137}$	-
Minimum ISE murill	$\frac{1.495}{K} \left[ \frac{T}{L} \right]^{0.945}$	$\frac{T}{1.101} \left[ \frac{L}{T} \right]^{0.771}$	$0.56T \left[ \frac{L}{T} \right]^{1.006}$	$0.1 < \frac{L}{T} < 1$

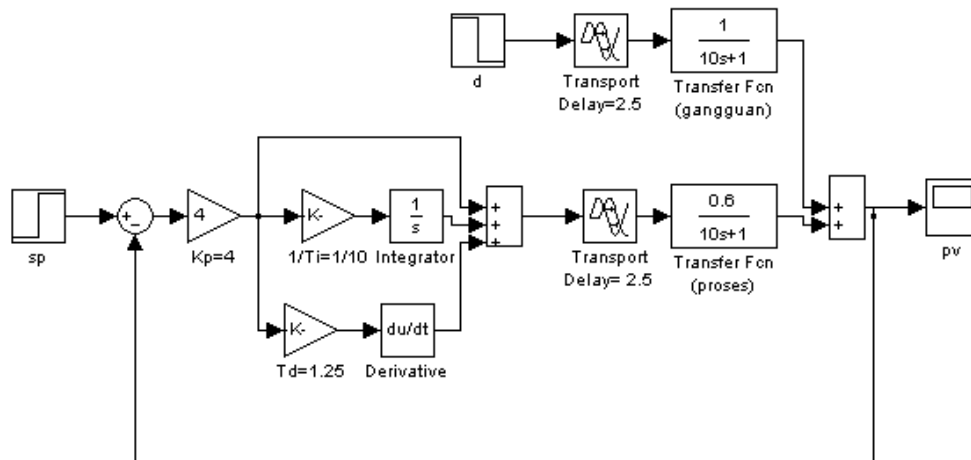
**Tabel 2.8.** Beberapa metode tuning kontrol PID ideal untuk proses IPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Cluet and Wang	$\frac{0.9588}{K^* L}$	$3.0425L$	$0.3912L$	Konstanta waktu loop tertutup= L
	$\frac{0.6232}{K^* L}$	$5.2586L$	$0.2632L$	Konstanta waktu loop tertutup= 2 L
	$\frac{0.466}{K^* L}$	$7.229L$	$0.2059L$	Konstanta waktu loop tertutup= 3 L
	$\frac{0.3752}{K^* L}$	$9.1925L$	$0.1702L$	Konstanta waktu loop tertutup= 4 L
Astrom-Haglung	$\frac{0.94}{K^* L}$	$2L$	$0.5L$	Ekuivalen Ultimated cycle ZN
Ford	$\frac{1.48}{K^* L}$	$2L$	$0.37L$	Quarter decay ratio
Rotach	$\frac{1.21}{K^* L}$	$1.6L$	$0.48L$	Faktor redaman terhadap gangguan =0.75

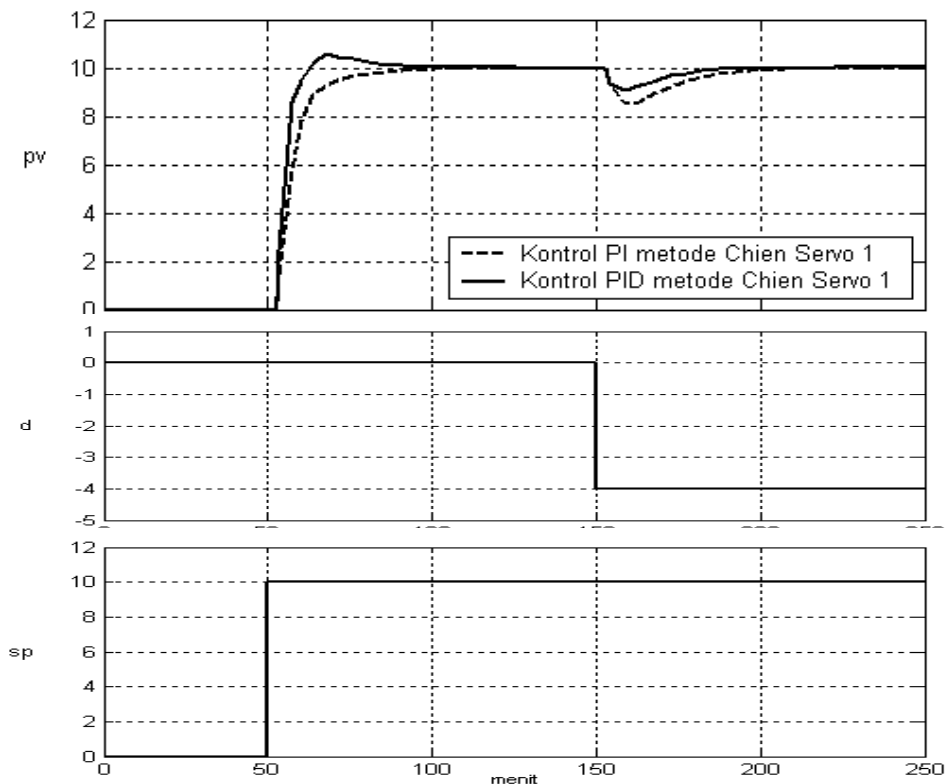
Seperti halnya pemilihan metode tuning kontrol PI, Pemilihan metode tuning kontrol PID juga sangat ditentukan oleh problem kontrol yang dihadapi serta unjuk kerja yang diharapkan dari sistem umpan baliknya. Sebagai contoh, jika untuk proses tertentu tidak ditolerir adanya overshoot pada respon output proses, maka sebagai tuning awal sebaiknya digunakan metode yang memang dikhususkan untuk itu --lihat keterangan tabel (Tetapi walaupun demikian, respon output proses sebenarnya secara praktis mungkin tidak akan benar-benar seperti apa yang tertulis pada keterangan tabel tersebut, - lihat penjelasannya pada bagian Mode Kontrol Proporsional Integral).

Seperti dua kasus pengontrolan sebelumnya, untuk melihat unjuk kerja dan dinamika sistem kontrol PID, disini akan digunakan tool Simulink untuk simulasinya. Agar terjadi kesinambungan dalam analisis, kita tetap jadikan sistem *Heat Exchanger* pada persamaan 2.8 sebagai bahan studi kasus pengontrolan.

Gambar 2.23 dan 2.24 berikut berturut-turut memperlihatkan diagram blok Simulink kontrol PID untuk proses *Heat Exchanger* tersebut serta salah satu contoh hasil simulasinya (Dalam simulasi ini parameter PID dicari dengan menggunakan metode PID Chien-servo 1, dan sebagai bahan perbandingan, pada grafik output yang sama diplot juga hasil pengontrolan PI dengan metode tuning dari Chien-servo 1 sebelumnya).



**Gambar 2.23.** Diagram blok simulink pengontrolan sistem Heat Exchanger dengan kontrol PID



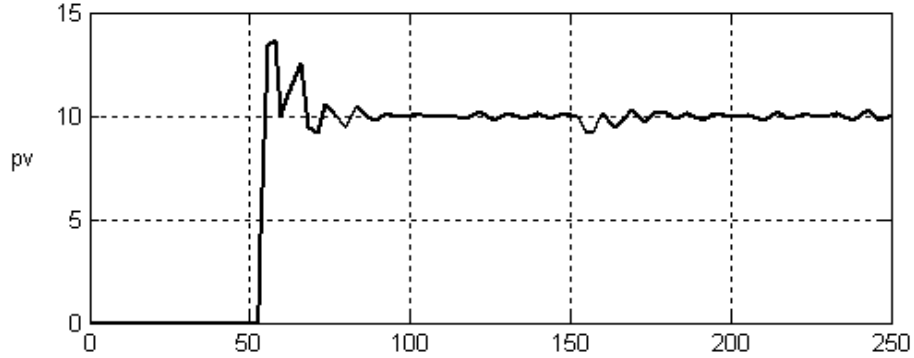
**Gambar 2.24.** Salah satu hasil simulasi simulink gambar 2.23

Sebagaimana telah dijelaskan, penyisipan derivative dalam sebuah kontrol PID pada dasarnya difungsikan untuk memperbaiki atau mempercepat respon transien output, hal ini sangat nampak dari hasil perbandingan simulasi kontrol PI dan PID dalam gambar 2.24 diatas.

Tetapi perbaikan respon itu sendiri pada kenyataannya sangat ditentukan juga oleh metode tuning yang digunakan. Untuk metode lain yang dipilih, mungkin saja penyisipan derivator ini tidak akan meningkatkan unjuk kerja sistem kontrol, bahkan secara praktis bisa jadi sistem menjadi kurang stabil. Hal ini seperti nampak dari hasil simulasi



Simulink gambar 2.25 dibawah (dalam gambar tersebut, parameter PID dicari dengan menggunakan metode Ziegler-Nichols) .



**Gambar 2.25.** Hasil simulasi kontrol PID dengan metode tuning Ziegler Nichols

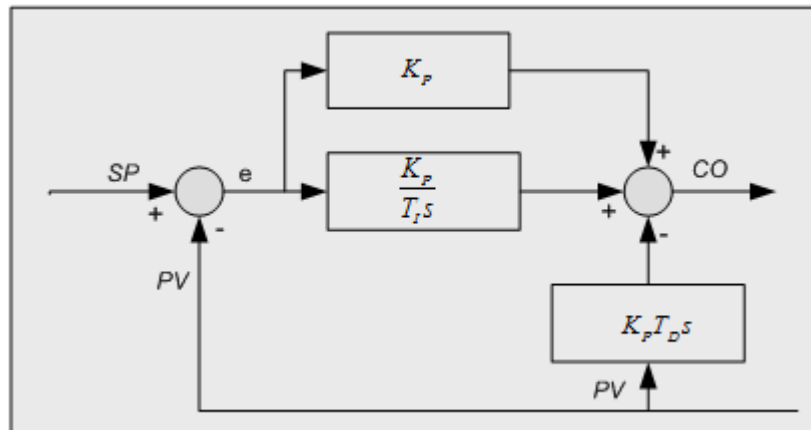
### 2.2.2 Struktur PID Tipe B

Dalam sejumlah modul PID komersil, Struktur PID tipe B dikenal juga dengan nama PID jenis PI-D. Beberapa vendor kontrol PID yang menggunakan algoritma jenis ini diantaranya adalah Allen Bradley dan Emerson. Perbedaan utama antara PID tipe B dengan struktur PID tipe A (PID ideal) hanya terletak pada sinyal input suku derivative-nya saja. Dalam hal ini sinyal input yang diderivative-kan pada PID tipe B bukan sinyal error, melainkan output proses (PV) secara langsung (lihat persamaan 2.12 dan diagram bloknya pada gambar 2.26).

$$CO(t) = K_p \left[ e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau - T_D \frac{dPV(t)}{dt} \right] \quad (2.12)$$

Dalam kawasan Laplace persamaan (2.12) dapat ditulis:

$$CO(s) = K_p \left[ 1 + \frac{1}{T_I s} \right] e(s) - K_p T_D s PV(s) \quad (2.13)$$



**Gambar 2.26.** Diagram blok kontroler PI-D

Secara praktis, Struktur kontrol Tipe B ini digunakan untuk menghindari sinyal kontrol keluaran derivative yang sangat besar (*impuls*) seperti yang terjadi pada PID ideal sesaat setelah terjadi perubahan Setting Point (*SP*) yang mungkin diberikan (misal pada problem Servo).

Tetapi walaupun demikian, jika yang terjadi adalah perubahan beban atau gangguan, struktur PI-D ini tetap akan menghasilkan perubahan output kontroler yang sangat besar.

Ditinjau dari aspek tuning parameter kontrolnya, kita secara praktis dapat menggunakan metode-metode tuning PID ideal seperti yang tercantum pada tabel 2.7 (untuk proses FOPDT) dan tabel 2.8 (untuk proses IPDT), atau dapat juga menggunakan metode tuning yang memang diperuntukan khusus untuk struktur PI-D ini (lihat tabel 2.9 dan 2.10 dibawah)

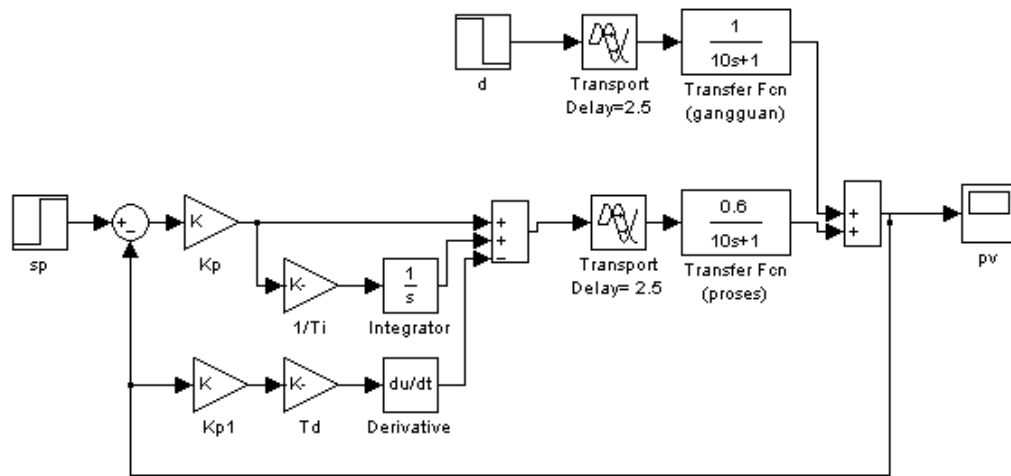
**Tabel 2.9.** Beberapa metode tuning kontrol PI-D untuk proses FOPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Minimum IAE Shinkey	$\frac{1.32T}{K.L}$	$1.80L$	$0.44L$	$\frac{L}{T} = 0.1$
	$\frac{1.32T}{K.L}$	$1.77L$	$0.41L$	$\frac{L}{T} = 0.2$
	$\frac{1.35T}{K.L}$	$1.43L$	$0.41L$	$\frac{L}{T} = 0.5$
	$\frac{1.49T}{K.L}$	$1.17L$	$0.37L$	$\frac{L}{T} = 1$
	$\frac{1.82T}{K.L}$	$0.92L$	$0.32L$	$\frac{L}{T} = 2$
Process Reaction Van Doren	$\frac{1.5T}{K.L}$	$2.5L$	$0.4L$	-

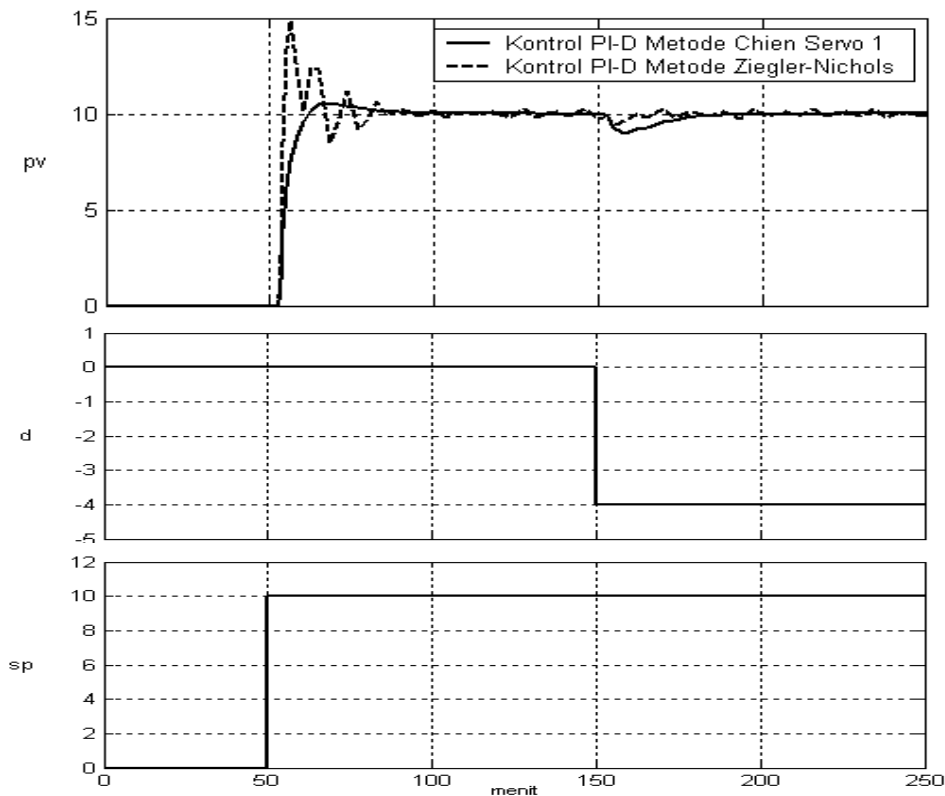
**Tabel 2.10.** Salah satu Metode tuning kontrol PI-D untuk proses IPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Minimum IAE Shinkey	$\frac{1.2821}{K^*.L}$	$1.90L$	$0.46L$	-

Untuk sebuah metode tuning PID ideal yang digunakan, respon proses yang dihasilkan oleh kontrol PI-D akan menunjukkan dinamika relative sama dengan respon sistem kontrol PID idealnya itu sendiri. Hal tersebut dapat dibuktikan secara cepat lewat uji simulasi. Gambar 2.27 dan 2.28 berikut berturut-turut memperlihatkan blok simulink kontrol PI-D pada sistem Heat Exchanger persamaan 2.8 dan salah satu hasil simulasinya (Dalam simulasi ini parameter PI-D dicari dengan menggunakan metode tuning PID Chien-servo 1, dan metode tuning PID Ziegler-Nichols yang tersaji pada tabel 2.7). .



**Gambar 2.27.** Diagram blok simulink pengontrolan sistem Heat Exchanger dengan kontrol PI-D



**Gambar 2.28.** Hasil simulasi untuk gambar 2.27

Berdasarkan hasil simulasi diatas, nampak bahwa tidak ada perbedaan yang mencolok antara respon kontrol PID dan PI-D untuk metode tuning yang sama. (bandingkan grafik hasil simulasi tersebut dengan gambar 2.24 dan 2.25).

### 2.2.3 Struktur PID Tipe C

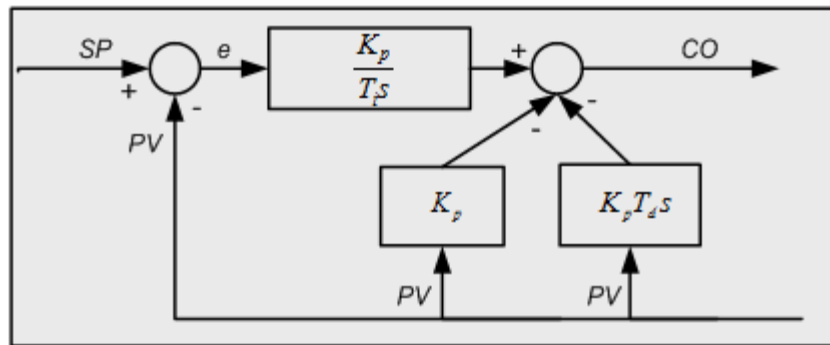
Dalam sejumlah buku teks kontrol, struktur PID tipe C dikenal juga dengan nama struktur I-PD (lihat persamaan 2.14 dan diagram bloknya pada gambar 2.28). Struktur kontrol PID tipe ini dapat dijumpai pada salah satu modul kontrol PID komersil produk

Emerson. Walaupun jarang digunakan tetapi struktur ini memiliki keunggulan yaitu menghilangkan sama sekali diskontinuitas pada keluaran sinyal kontrol akibat perubahan SetPoint ( $SP$ ). Dengan menggunakan struktur kontrol ini ketika nilai  $SP$  berubah, maka sinyal keluaran kontroler akan berubah secara halus.

$$CO(t) = K_p \left[ -PV(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau - T_D \frac{dPV(t)}{dt} \right] \quad (2.14)$$

Dalam kawasan laplace persamaan 2.14 diatas dapat ditulis:

$$CO(s) = K_p \left[ -PV(s) + \frac{e(s)}{T_I s} - T_D .s.PV(s) \right] \quad (2.15)$$



**Gambar 2.28.** Diagram blok kontroler I-PD

Berdasarkan gambar 2.28 tersebut, satu-satunya komponen kontrol yang bekerja pada error adalah blok integrator. Ketika terjadi error tangga yang disebabkan perubahan setpoint ( $SP$ ), maka output integrator akan berubah secara perlahan-lahan, di lain pihak output penguat proporsional dan derivative juga akan berubah secara halus, sehingga dengan demikian output kontrol totalnya tidak berubah secara agresive.

Walaupun demikian, seperti halnya struktur kontrol PI-D, struktur I-PD ini secara praktis tetap akan menghasilkan perubahan output kontrol yang sangat besar jika yang terjadi adalah perubahan beban atau gangguan.

Terkait dengan metode tuning-nya, tabel 2.11 dan 2.12 dibawah berturut-turut dapat digunakan sebagai basis perhitungan parameter kontrol I-PD untuk model proses FOPDT dan metode IPDT. Tetapi seperti halnya struktur PI-D, kita juga secara praktis dapat menggunakan metode tuning PID ideal (tabel 2.7 dan 2.8) untuk mendapatkan parameter-parameter kontroler I-PD ini.

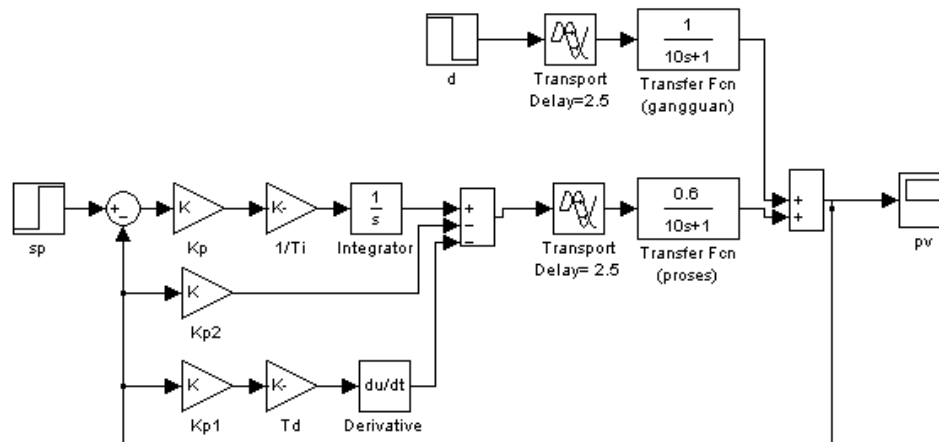
**Tabel 2.11.** Salah satu metode tuning kontrol PID tipe C untuk proses FOPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Minimum ISE -Argelaguete	$\frac{2T + L}{2K.L}$	$T + 0.5L$	$\frac{TL}{2T + L}$	-

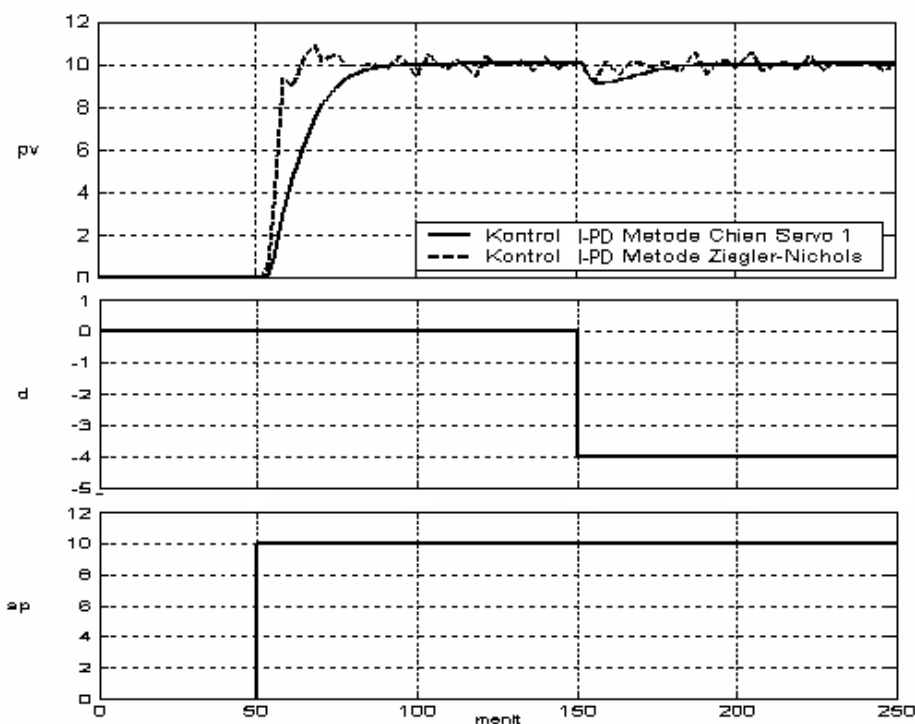
**Tabel 2.12.** Salah satu metode tuning kontrol PID tipe C untuk proses IPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Minimum ISE -	$\frac{1.414T_{cl} + L}{K^* (T_{cl}^2 + 0.707T_{cl}L + 0.25L^2)}$	$1.414T_{cl} + L$	$\frac{0.25L^2 + 0.707T_{cl}L}{1.414T_{cl} + L}$	$L < 0.2T_{cl}$

Argelaguete				
-------------	--	--	--	--



**Gambar 2.29.** Diagram blok simulink pengontrolan sistem Heat Exchanger dengan kontrol PI-D

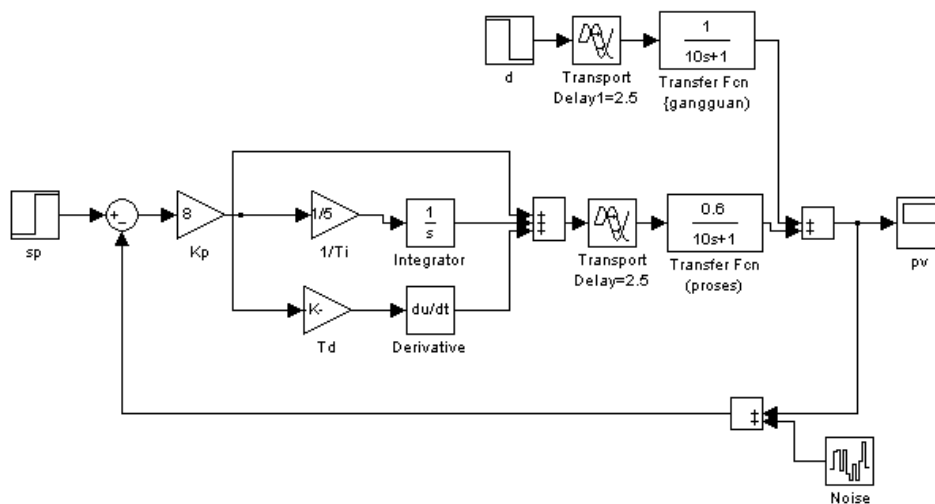


**Gambar 2.30.** Hasil simulasi untuk gambar 2.29

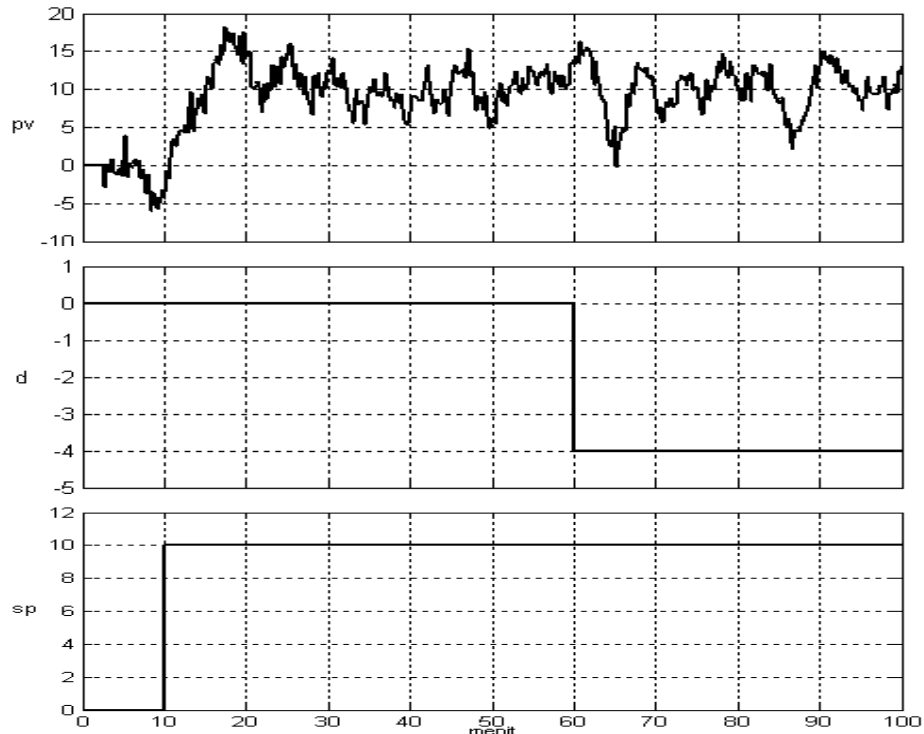
Tidak tergantung metode tuning yang digunakan, respon output proses pada saat menanggapi perubahan setpoint untuk struktur kontrol I-PD ini akan selalu menunjukkan keadaan transien yang relative lebih lambat dibandingkan jika metode tuning terkait diterapkan pada struktur PID ideal-nya. Gambar 2.29 dan 2.30 berturut-turut memperlihatkan blok simulink kontrol I-PD untuk sistem *Heat Exchanger* --persamaan 2.8 serta salah satu hasil simulasinya (Dalam simulasi tersebut parameter I-PD dicari dengan menggunakan metode tuning PID Chien-servo 1, dan metode tuning PID Ziegler-Nichols yang tersaji pada tabel 2.7).

### 2.2.4 Struktur kontrol PID dengan Filter (PIDF)

Selain akan menghasilkan sinyal kontrol yang sangat besar ketika terjadi perubahan Setpoint atau beban, Salah satu kelemahan lain dari penggunaan suku derivative pada pengontrolan PID (baik PID tipe A, B, maupun C) adalah diperkuatnya *noise* yang pasti muncul dalam pengukuran output proses (PV).



**Gambar 2.31.** Diagram blok simulink pengontrolan sistem Heat Exchanger dengan kontrol PID yang telah ditambahkan simulasi noise



**Gambar 2.32.** Hasil simulasi untuk gambar 2.31

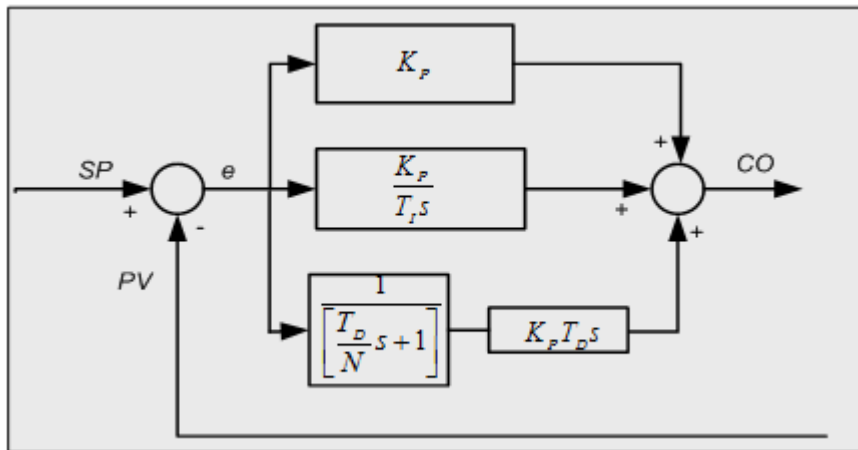
Untuk melihat pengaruh *noise* atau derau pengukuran, kita secara langsung dapat mengujinya lewat Simulink. Gambar 2.31 dan 2.32 berturut-turut memperlihatkan blok

kontrol PID ideal tanpa filter (yang didalamnya telah sisipkan sumber *noise*) serta salah satu grafik hasil simulasinya (dalam simulasi tersebut, parameter PID dicari dengan menggunakan metode tuning PID Ziegler-Nichols pada table 2.7)

Berdasarkan hasil simulasi tersebut, nampak bahwa dengan adanya derau pengukuran, respon output proses (*pv*) akan sedikit berosilasi disekitar nilai SetPointnya. Hal ini disebabkan derau yang muncul secara praktis akan diperkuat oleh penguat derivative (komponen derivative ini pada dasarnya adalah sebuah filter pelewat frekuensi tinggi).

Untuk menghindari hal tersebut, dalam sebagian besar modul PID komersil, komponen derivative pada kontroler umumnya di-*cascade* atau diserikan dengan sebuah filter orde satu (atau lebih). Persamaan 2.16 berikut ini memperlihatkan struktur PID ideal (tipe A) dalam kawasan Laplace yang telah dilengkapi filter orde satu (lihat juga blok diagramnya pada gambar 2.33):

$$CO(s) = K_p \left[ 1 + \frac{1}{T_i s} + \frac{T_D s}{\frac{T_D}{N} s + 1} \right] E(s) \quad (2.16)$$



**Gambar 2.33.** Diagram blok kontrol PID (ideal) dengan filter orde 1

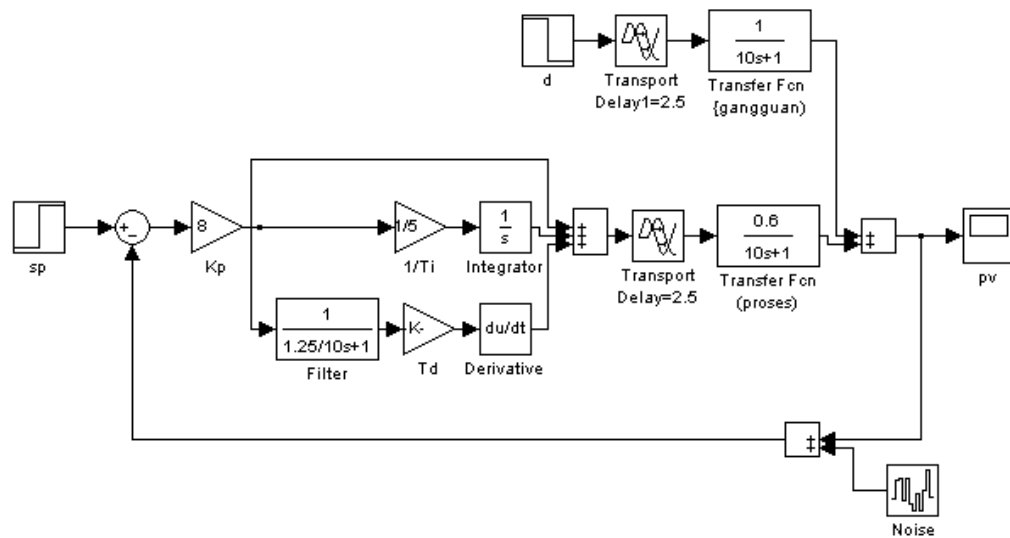
Dalam kawasan waktu nyata, persamaan 2.16 dapat ditulis:

$$CO(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_D \frac{de_f(t)}{dt} \right] \quad (2.17)$$

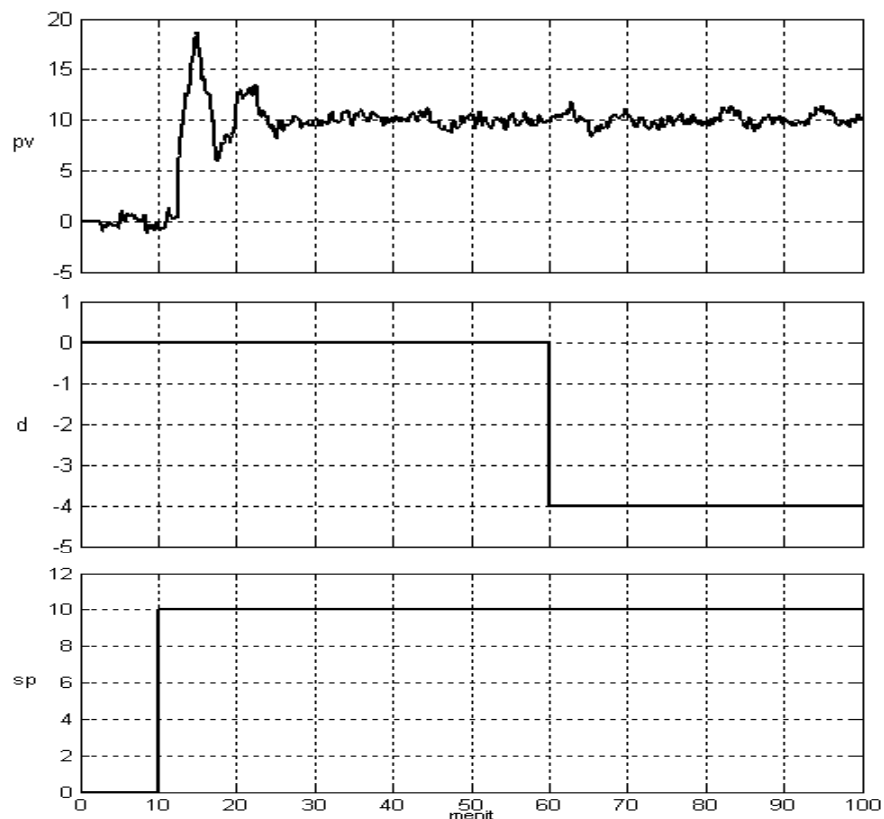
dalam hal ini:

$$\frac{T_D}{N} \frac{de_f(t)}{dt} + e_f(t) = e(t) \quad (2.18)$$

Besarnya N pada PIDF komersil berkisar antara 1 sampai 33, tetapi yang paling umum adalah antara 8 sampai 16. Nilai N ini pada dasarnya berfungsi untuk mengatur lebar pita filter orde satu yang digunakan. Semakin kecil nilai N, maka konstanta waktu filter akan semakin besar (lihat persamaan 2.16).



**Gambar 2.34.** Diagram blok simulink pengendalian sistem Heat Exchanger dengan kontrol PIDF



**Gambar 2.35.** Hasil simulasi untuk gambar 2.34

Besar kecilnya konstanta waktu filter tersebut akan mempengaruhi pelemahan dari noise frekuensi tinggi yang muncul pada pengukuran: Semakin besar konstanta waktu yang digunakan, pelemahan noise akan semakin besar (tapi dilain pihak respon output kontrolnya pun akan semakin lambat).



Untuk melihat pengaruh penambahan filter pada dinamika output proses, disini kita akan gunakan blok simulink untuk proses *heat exchanger* sebelumnya. Gambar 2.34 dan 2.35 berturut-turut memperlihatkan blok simulink kontrol PID ideal yang telah dilengkapi filter orde 1 (dengan  $N=10$ ) beserta salah satu grafik hasil simulasinya (seperti halnya simulasi terdahulu, pada simulasi ini metode tuning yang digunakan adalah metode tuning PID ideal Ziegler-Nichols).

Seperti nampak dari hasil simulasi gambar 2.35, dengan adanya filter pelewat frekuensi rendah orde 1 yang disisipkan pada suku kontrol derivative, output prosesnya relative stabil jika dibandingkan dengan hasil pengontrolan PID tanpa filter (bandingkan gambar tersebut dengan gambar 2.32)

Selain tabel 2.7 dan tabel 2.8, khusus untuk struktur kontrol PIDF, tabel 2.13 dan 2.14 dibawah berturut-turut dapat digunakan sebagai basis penentuan parameter PIDF awal untuk model proses FOPDT dan IPDT.

**Tabel 2.13.** Beberapa metode tuning kontrol PIDF untuk proses FOPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Chien	$\frac{T + 0.5L}{K(\lambda + 0.5L)}$	$T + 0.5L$	$\frac{TL}{2T + L}$	$\lambda = [L, T], N=10$
Davydov	$\frac{1}{K \left[ 1.552 \frac{L}{T} + 0.078 \right]}$	$\left[ 0.186 \frac{L}{T} + 0.532 \right]$	$0.25 \left[ 0.186 \frac{L}{T} + 0.532 \right]$	Closed loop respon, faktor redaman=0.9, $0.2 \leq \frac{L}{T} \leq 1, N = K$

**Tabel 2.14.** metode tuning kontrol PIDF untuk proses IPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Chien	$\frac{2}{K^*(\lambda + 0.5L)}$	$2\lambda + T$	$\frac{L(\lambda + 0.25L)}{2\lambda + L}$	$\lambda = \frac{1}{K^*}, N = 10, N=10$

### 2.2.5 Struktur Kontrol PID Standar ISA dan PID Bentuk Umum

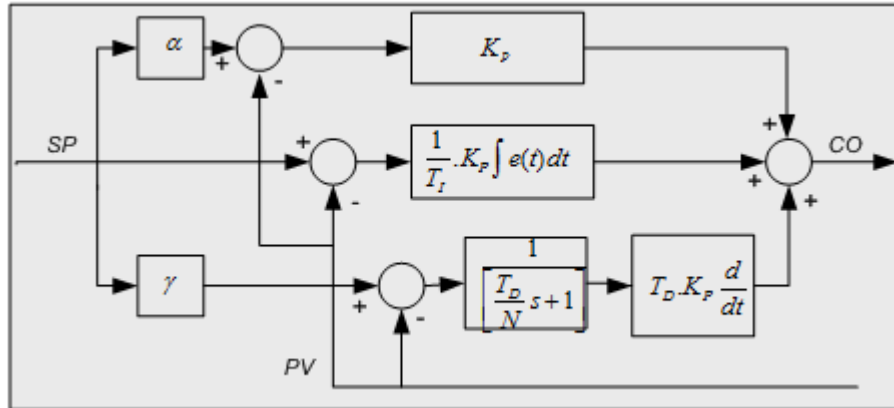
Persamaan (2.19) berikut ini adalah persamaan untuk PID yang dikenal dengan nama struktur PID standar ISA

$$CO(s) = K_P \left[ (\alpha SP(s) - PV(s)) + \frac{E(s)}{T_I s} + \frac{T_D s}{\frac{T_D}{N} s + 1} [\gamma SP(s) - PV(s)] \right] \quad (2.19)$$

atau dapat juga ditulis:

$$CO(S) = K_P (\alpha SP(s) - PV(s)) + \frac{K_P E(s)}{T_I s} + \frac{K_P T_D s}{\frac{T_D}{N} s + 1} [\gamma SP(s) - PV(s)] \quad (2.20)$$

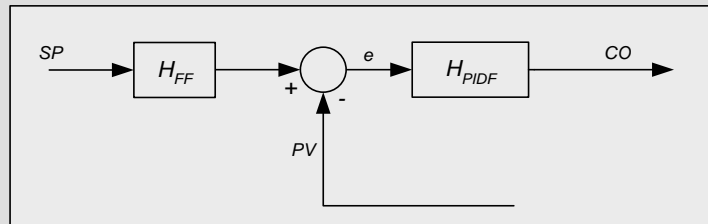
Dalam hal ini,  $\alpha$  dan  $\gamma$  adalah faktor-faktor pembobot yang besarnya dapat dipilih secara bebas. Dalam bentuk diagram bloknya, persamaan 2.19 dapat digambarkan seperti nampak pada gambar 2.36.



**Gambar 2.36.** Diagram blok pengontrolan PID standar ISA

#### Catatan Penting Tentang PID Struktur ISA

Secara teknis, struktur PID ISA ini dapat dipandang sebagai sebuah struktur PID yang dikenal dengan nama struktur PID dengan 2 derajat kebebasan (Two degrees of freedom: dof) seperti nampak pada gambar 2.37 dibawah.



**Gambar 2.37.** Diagram blok PID 2 derajat kebebasan

Dalam hal ini  $H_{PIDF}$  adalah fungsi alih PID ideal yang dilengkapi filter orde 1 pada suku derivative-nya:

$$H_{PIDF}(s) = K_P \left[ 1 + \frac{1}{T_I s} + \frac{T_D s}{\frac{T_D}{N} s + 1} \right] \quad (2.21)$$

Sedangkan  $H_{FF}$  adalah fungsi alih yang memiliki bentuk seperti pada persamaan 2.22.

$$H_{FF} = \frac{1 + s \left[ \alpha T_I + \frac{T_D}{N} \right] + s^2 T_I T_D \left[ \beta + \frac{\alpha}{N} \right]}{1 + s \left[ T_I + \frac{T_D}{N} \right] + s^2 T_I T_D \left[ 1 + \frac{1}{N} \right]} \quad (2.22)$$

Secara fungsional, fungsi alih  $H_{PIDF}$  berlaku sebagai sistem regulator (sehingga parameter kontrolnya secara praktis dapat dicari dengan menggunakan metode-metode tuning untuk kasus tersebut). Sedangkan  $H_{FF}$  dalam hal ini berlaku sebagai filter yang digunakan untuk meningkatkan performansi sistem servo --SetPoint Tracking (dengan alasan ini maka struktur kontrol pada gambar 2.37 dinamakan struktur PID dengan 2 derajat kebebasan).

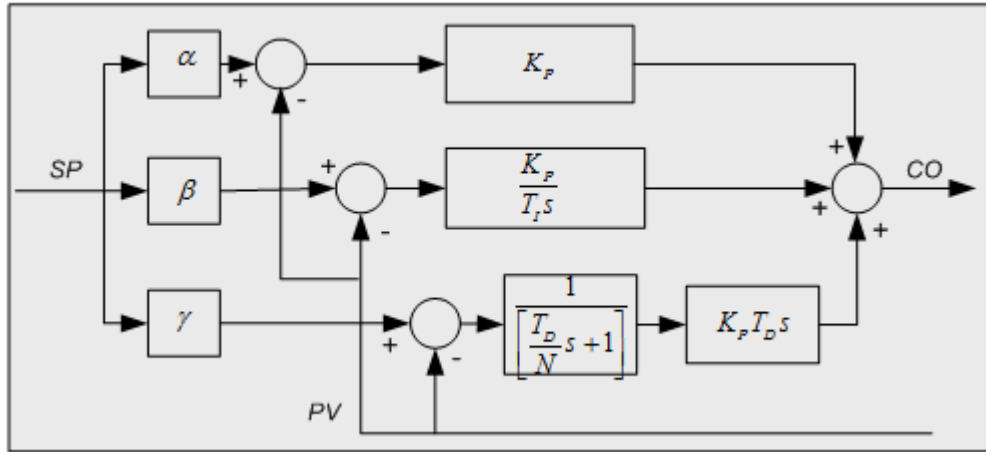
sebelumnya. Sebagai contoh jika  $\alpha$  dan  $\gamma$  berturut-turut dipilih sama dengan satu, maka struktur akhir yang didapat adalah struktur PID ideal (dengan filter), jika  $\alpha$  dan  $\gamma$  berturut-turut dipilih sama dengan satu dan nol, maka struktur akhir yang didapat adalah

struktur PID tipe B (dengan filter), sedangkan jika  $\alpha$  dan  $\gamma$  berturut-turut dipilih sama dengan nol, maka struktur akhir yang didapat adalah struktur PID tipe C (dengan filter).

Tidak hanya SetPoint ( $SP$ ) untuk blok proporsional dan diferensiator saja yang dapat diberikan faktor pembobot, sinyal  $SP$  untuk blok integrator pun pada dasarnya dapat diberikan faktor pembobot yang besarnya dapat di-setting secara bebas seperti yang diperlihatkan pada persamaan 2.23 (hanya saja jika hal tersebut dilakukan, maka tuning parameternya secara praktis akan semakin sulit).

$$CO(s) = K_p \left[ (\alpha SP(s) - PV(s)) + \frac{(\beta SP(s) - PV(s))}{T_i s} + \frac{T_d s}{\frac{T_d}{N} s + 1} [\gamma SP(s) - PV(s)] \right] \quad (2.23)$$

Dalam literature kontrol, struktur kontrol PID dengan bentuk persamaan 2.23 tersebut dikenal dengan nama struktur PID bentuk umum (*general*). Blok diagram untuk struktur ini diperlihatkan oleh gambar 2.38 berikut:



Gambar 2.38. Diagram blok pengontrolan PID bentuk general

### 2.2.6. Struktur PID Serial

Tidak seperti struktur-struktur PID yang telah dijelaskan sebelumnya, struktur PID serial ini terdiri dari komponen-komponen kontrol yang disusun secara serial. Berdasarkan kenyataan di lapangan, dapat dikatakan bahwa struktur PID serial kurang begitu populer dibandingkan struktur-struktur PID jenis paralel.

Walaupun demikian, beberapa modul kontroler PID komersil masih menawarkan struktur serial sebagai salah satu basis algoritma kontrolnya (salah satu diantaranya dapat dijumpai pada modul kontrol PID komersil produk Foxboro).

Persamaan (2.24) berikut memperlihatkan struktur PID serial:

$$CO(t) = K_p [e_1(t) + \frac{1}{T_i} \int e_1(t) dt] \quad (2.24)$$

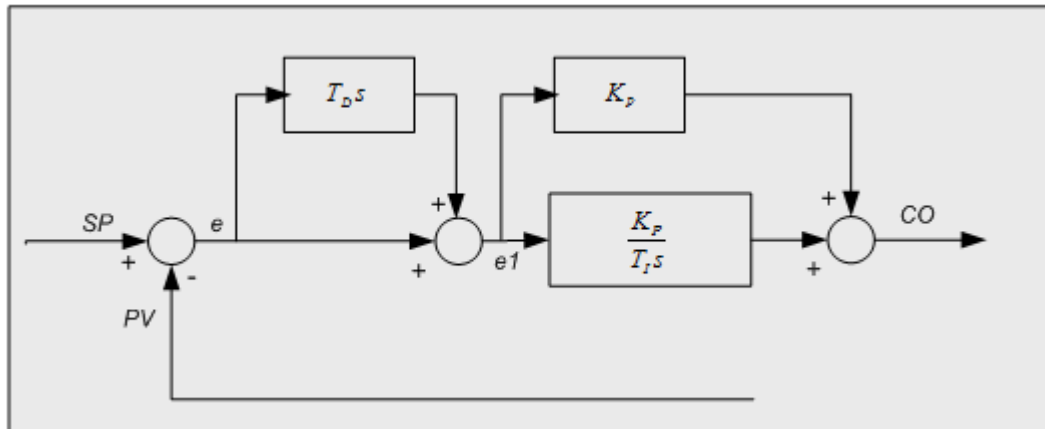
dengan:  $e_1(t) = e(t) + T_d \frac{de(t)}{dt}$

Dalam kawasan laplace, struktur PID serial dapat dituliskan seperti pada persamaan (2.25) berikut:

$$CO(s) = K_p \left(1 + \frac{1}{T_I s}\right) (1 + T_D s) e(s) \quad (2.25)$$

Berdasarkan persamaan (2.25), terlihat bahwa jika penguat diferensial ( $T_D$ ) di-setting sama dengan nol, maka persamaan (2.25) diatas akan memiliki relasi yang sama dengan sistem kontrol PI regular (lihat **subbab 2.2.1**).

Jika digambarkan dalam bentuk diagram blok-nya, PID serial ini akan nampak seperti pada gambar 2.39 dibawah



**Gambar 2.39.** Blok pengontrolan PID serial

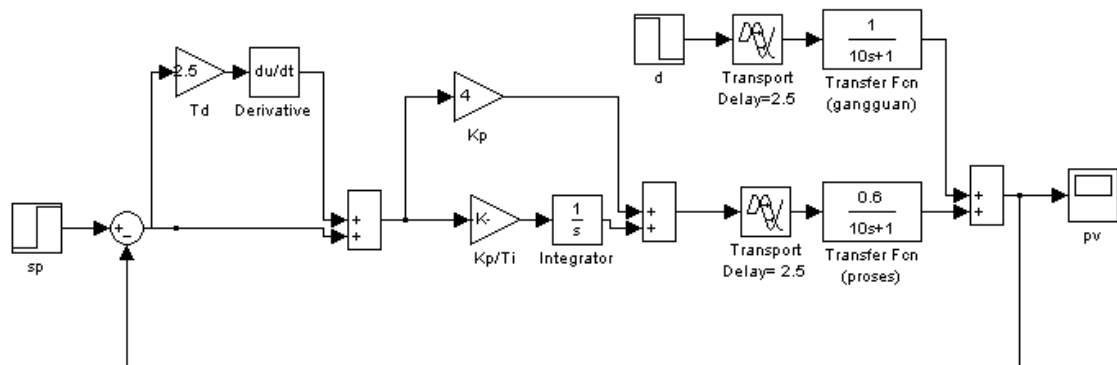
Karena struktur kontrol PID serial relative berbeda dengan struktur-struktur kontrol PID jenis paralel, maka terkait dengan masalah tuning parameter kontrolnya, kita secara praktis tidak dapat menggunakan metode-metode tuning yang telah disajikan sebelumnya.

Khusus untuk PID serial, terdapat dua metode tuning populer yang dapat digunakan sebagai tuning awal kontrol proses FOPDT: Metode Ziegler-Nichols dan Astrom-Haglung (lihat tabel 2.15)

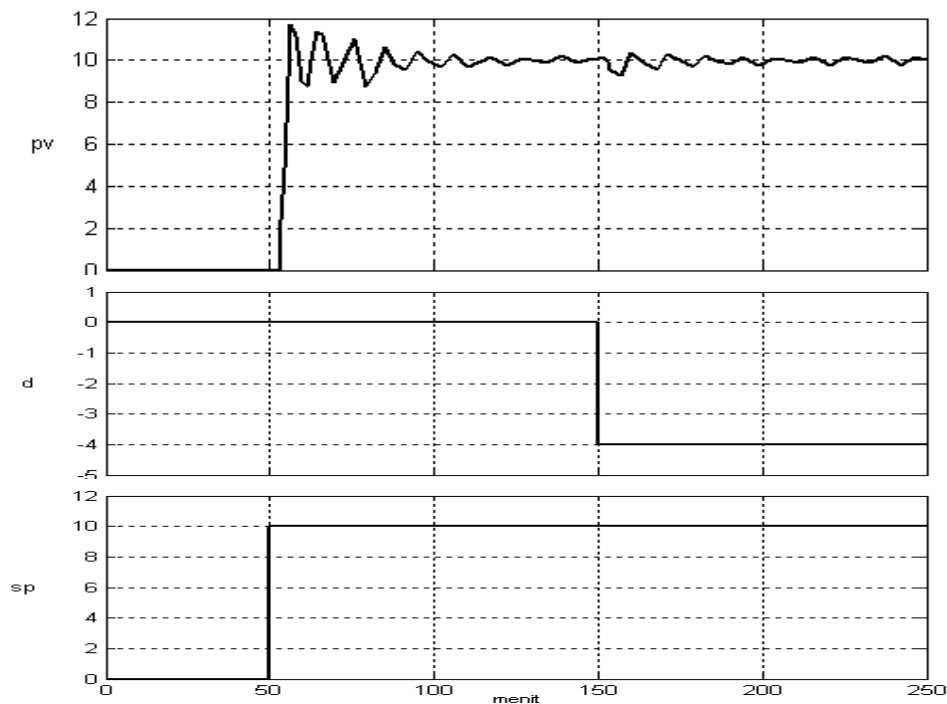
**Tabel 2.15.** Beberapa metode tuning kontrol PID serial untuk proses FOPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Ziegler-Nichols	$\frac{0.6T}{K.L}$	$L$	$L$	-
Astrom-Haglung	$\frac{5T}{6K.L}$	$1.5L$	$0.25L$	Foxboro EXACT

Seperti kasus-kasus sebelumnya, untuk melihat unjuk kerja kontrol PID serial, disini akan digunakan Simulink pada pengontrolan model proses Heat-Exchanger -- persamaan 2.8. Gambar 2.40 dan 2.41 berikut berturut-turut memperlihatkan blok Simulink kontrol PID Serial beserta salah satu grafik hasil simulasinya (dalam simulasi ini metode tuning yang digunakan adalah metode Ziegler-Nichols yang tersaji pada tabel 2.15).



**Gambar 2.40.** Diagram blok simulink pengontrolan sistem Heat Exchanger dengan kontrol PID Serial



**Gambar 2.41.** Hasil simulasi untuk gambar 2.40

Berdasarkan hasil simulasi tersebut, nampak bahwa untuk model proses yang diberikan, unjuk kerja kontrol PID ideal dengan menggunakan metode tuning Ziegler-Nichols ini menghasilkan kestabilan yang relative kurang pada saat terjadi perubahan setpoint maupun saat terjadi gangguan. (tetapi overshoot yang dihasilkan nilainya relative kecil).

Seperti yang telah sedikit disinggung sebelumnya, terkait dengan masalah tuning, walaupun secara praktis metode-metode tuning PID paralel tidak dapat diterapkan pada PID serial (dan begitu juga sebaliknya), tetapi jika untuk kasus tertentu, nilai parameter PID serial optimal telah diketahui, maka parameter PID paralel (ideal) –nya dapat dicari berdasarkan relasi dibawah ini:

$$K_{PP} = K_{PS} \frac{T_{IS} + T_{DS}}{T_{IS}} \quad (2.26)$$

---

$$T_{IP} = T_{IS} + T_{DS} \quad (2.27)$$

$$T_{DP} = \frac{T_{IS} T_{DS}}{T_{IS} + T_{DS}} \quad (2.28)$$

dengan  $K_{PP}$  = Gain proporsional (pada PID paralel)

$T_{IP}$  = Waktu integral (pada PID paralel)

$T_{DP}$  = Waktu derivative (pada PID paralel)

$K_{PS}$  = Penguat proporsional (pada PID serial)

$T_{IS}$  = Waktu integral (pada PID serial)

$T_{DS}$  = Waktu derivative (pada PID serial)

Sedangkan jika yang diketahui adalah nilai parameter-parameter PID paralel (ideal), maka parameter PID serial-nya dapat dicari dengan menggunakan relasi berikut:

$$K_{PS} = \frac{K_{PP}}{2} \left[ 1 + \sqrt{1 - \frac{4T_{DP}}{T_{IP}}} \right] \quad (2.29)$$

$$T_{IS} = \frac{T_{IP}}{2} \left[ 1 + \sqrt{1 - \frac{4T_{DP}}{T_{IP}}} \right] \quad (2.30)$$

$$T_{DS} = \frac{T_{IP}}{2} \left[ 1 - \sqrt{1 - \frac{4T_{DP}}{T_{IP}}} \right] \quad (2.31)$$

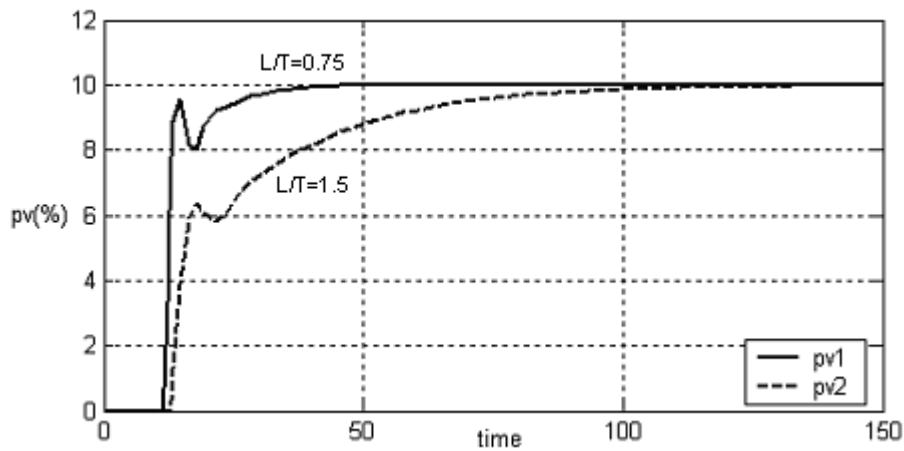
Tetapi satu hal penting yang perlu diperhatikan, tiga relasi terakhir ini hanya berlaku jika

$$T_{IP} > T_{DP}.$$

### 2.3. Struktur Kontrol PID dengan Kompensator Delay: Smith Predictore dan Predictive Proporsional Integral (pPI)

Tidak tergantung dari struktur dan metode tuning yang digunakan, penerapan Kontrol PID pada proses-proses yang memiliki karakteristik keterlambatan waktu yang sangat dominan ( $L/T > 1$ ) secara praktis akan selalu menghasilkan unjukkerja yang tidak memuaskan. Semakin besar  $L/T$  maka unjuk kerja sistem kontrol PID akan semakin menurun.

Untuk sebuah struktur dan metode tuning yang dipilih, respon proses akan semakin lambat seiring dengan membesarnya perbandingan  $L/T$  tersebut. Gambar 2.42 dibawah memperlihatkan hasil simulasi kontrol PID ideal dengan menggunakan metode tuning Ziegler-Nichols untuk dua buah proses yang masing-masing memiliki perbandingan  $L/T=0.75$  dan  $L/T=1.5$ .

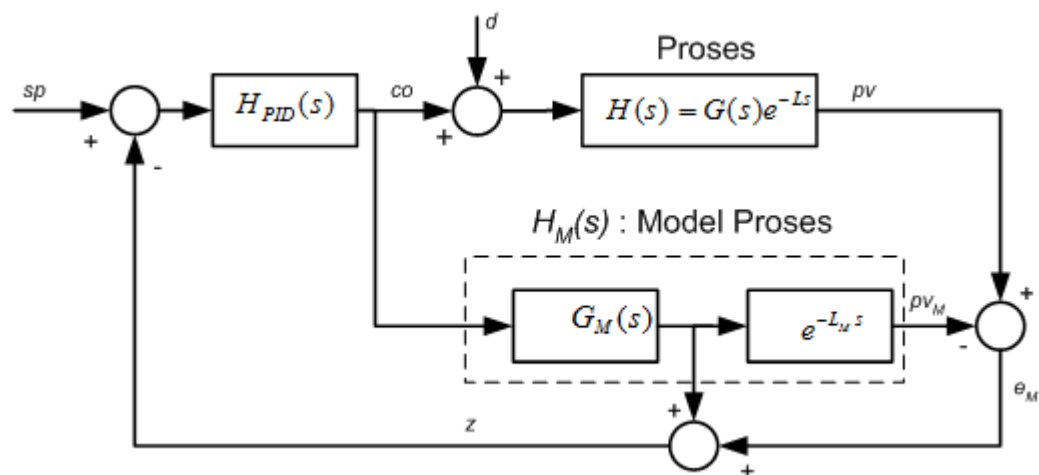


**Gambar 2.42** Hasil simulasi kontrol PID ideal untuk dua buah proses dengan  $L/T$  yang berbeda

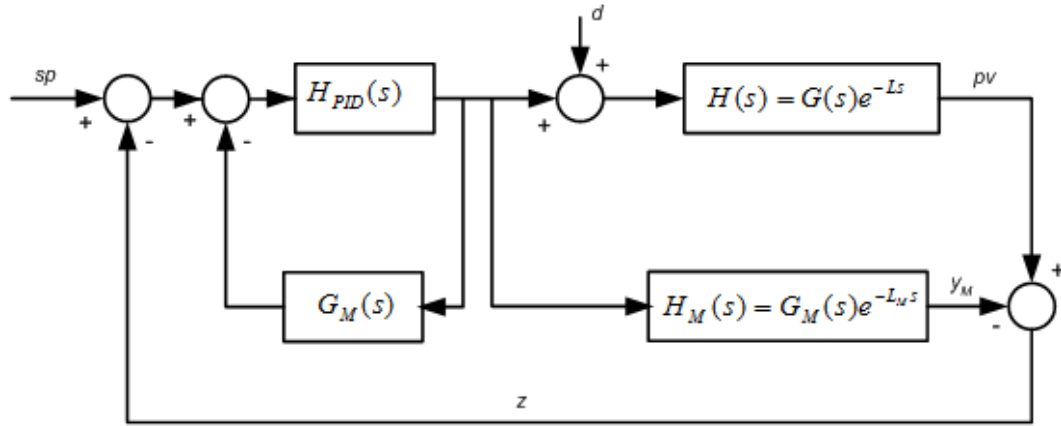
Seperti nampak dari hasil simulasi tersebut, tanggapan proses yang memiliki  $L/T$  relative besar (1.5) secara praktis akan menghasilkan respon output sangat lambat dibandingkan dengan proses yang memiliki perbandingan  $L/T$  relative kecil (0.75)

Secara praktis, respon proses yang lambat tersebut dapat saja dibuat relative lebih cepat yaitu dengan cara memperbesar nilai  $K_p$  dan/atau memperkecil  $T_i$  secara manual oleh operator. Tetapi satu hal yang perlu diperhatikan, perubahan nilai gain tidak boleh terlalu besar dari nilai hasil perhitungan, hal ini dikarenakan respon proses nantinya akan menjadi kurang bahkan tidak stabil (terlalu berosilasi).

Untuk mengatasi problem keterlambatan transportasi tersebut, sekitar tahun 1957 Smith mengusulkan sebuah struktur kontrol PID yang sekarang dikenal dengan nama Smith Predictore (lihat gambar 2.43 dan 2.44).



**Gambar 2.43** Struktur kontrol Smith Predictore



**Gambar 2.44** Diagram blok alternative Struktur kontrol Smith Predictore

Salah satu perbedaan Smith predictore dengan struktur kontrol PID biasa adalah digunakannya model matematis proses secara eksplisit dalam persamaan kontroler :

$$H_M(s) = G_M(s)e^{-L_M s} \quad (2.32)$$

Jika proses yang dikontrol memiliki karakteristik model FOPDT, maka persamaan 2.32 diatas dapat direpresentasikan menjadi:

$$H_M(s) = \frac{K_M}{T_M s + 1} e^{-L_M s} \quad (2.33)$$

Dalam hal ini besarnya parameter-parameter model proses tersebut ( $K_M$  dan  $T_M$ ) dapat dicari lewat identifikasi *Bump Test* seperti yang telah dijelaskan pada **Bab 1**.

Perhatikan, jika model proses hasil identifikasi betul-betul merepresentasikan proses sebenarnya:  $H_M(s)=H(s)$ , maka berdasarkan gambar 2.43, error model ( $e_M$ ) yang dihasilkan akan bernilai nol. Sehingga dengan demikian, ditinjau dari sudut pandang perancangan kontroler, parameter PID dapat dicari hanya berdasarkan informasi model  $G_M(s)$  yang teridentifikasi saja (perhatikan gambar 2.45).



**Gambar 2.45** Reduksi Struktur kontrol Smith Predictore untuk kasus  $H_M(s) = H(s)$

Untuk model proses FOPDT yang dikontrol,  $G_M(s)$  ini tidak lain adalah bagian fungsi alih yang bebas delay (atau sistem orde satu) :

$$G_M(s) = \frac{K_M}{T_M s + 1} \quad (2.34)$$



Khusus untuk pengontrolan proses atau sistem orde satu, struktur kontrol yang cocok digunakan adalah struktur kontrol PI, hal tersebut disebabkan selain karena kemudahan dalam mencari parameter kontrolnya, juga performansi sistem kontrol dapat diatur sesuai kehendak perancang.

Persamaan (2.35) berikut ini memperlihatkan hubungan parameter model proses orde satu dengan parameter kontrol PI yang harus diseting (lihat pembuktian matematisnya dalam **Subbab contoh soal dan penyelesaian**).

$$K_p = \frac{T_M}{K_M T_{cl}}; T_I = T_M \quad (2.35)$$

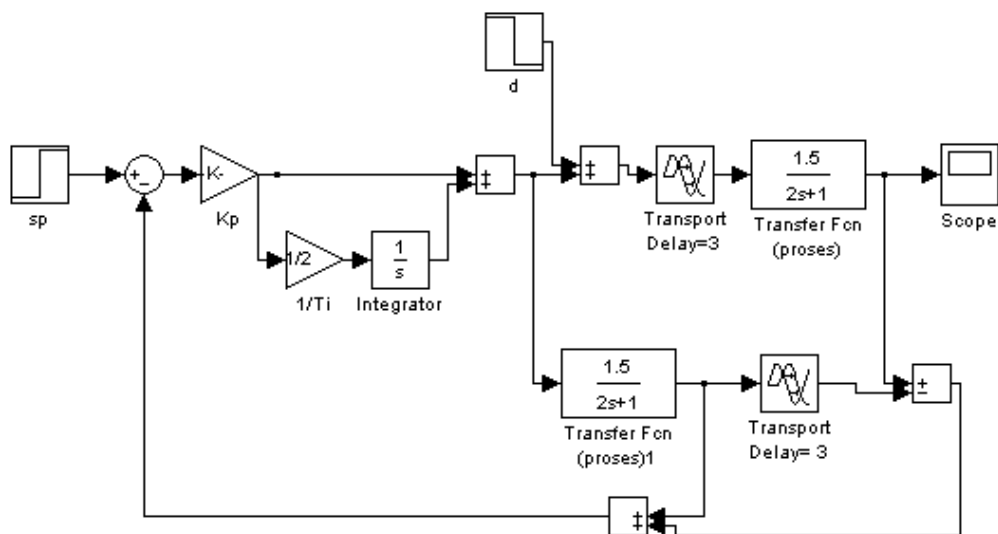
dengan :

$K_M$  = gain statis model proses

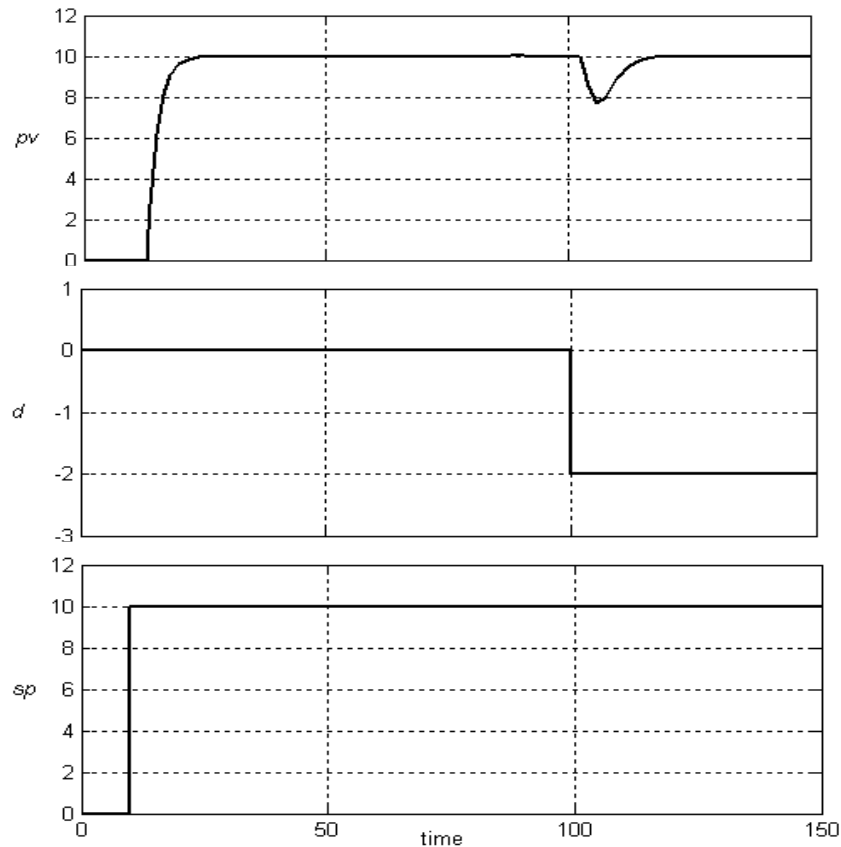
$T_M$  = Konstanta waktu model proses

$T_{cl}$  = Konstanta waktu umpan balik proses yang dikehendaki perancang

Agar mendapat pemahaman yang mendalam, sebaiknya performansi smith predictor tersebut diuji lewat simulasi. Gambar 2.46 dan 2.47 berikut memperlihatkan blok simulink untuk kontrol Smith Predicotre beserta salah satu hasil simulasinya (dalam simulasi ini kontroler yang digunakan adalah jenis PI dengan besar parameter kontrol dicari lewat persamaan 2.35. besar konstanta waktu umpan balik yang diinginkan  $T_{cl} = 2$ )



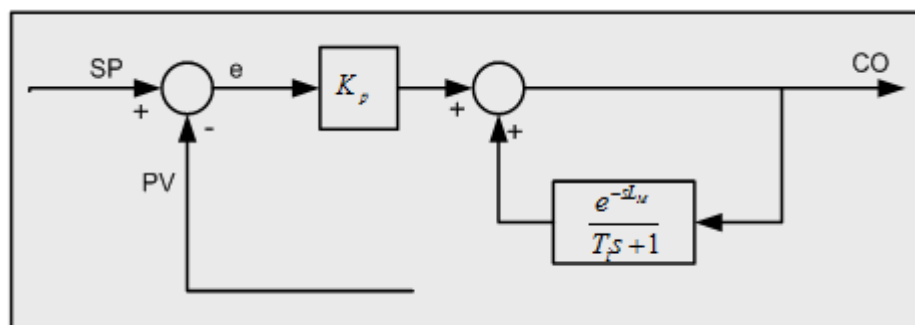
**Gambar 2.46** Blok Simulink untuk kontrol Smith Predictor



**Gambar 2.47** Hasil simulasi gambar 2.46

Salah satu bentuk khusus struktur kontrol Smith Predictor yang cukup populer adalah apa yang dinamakan dengan struktur kontrol Predictive PI (pPI). Struktur kontrol ini dapat dijumpai pada salah satu modul komersil produk ABB.

Dalam bentuk blok diagramnya, struktur kontrol pPI memiliki struktur yang hampir sama dengan bentuk kontrol PI alternatif pada gambar 2.13. Hanya saja dalam struktur kontrol Predictive PI ini, pada bagian umpan balik positive-nya telah ditambahkan model delay transportasi dari proses yang dikontrol (lihat gambar 2.48).



**Gambar 2.48** Diagram blok sistem kontrol Predictive PI

## 2.4. Dua Fitur Penting pada Sistem Kontrol PID Komersil: AntiWindUp dan Bumpless Transfer

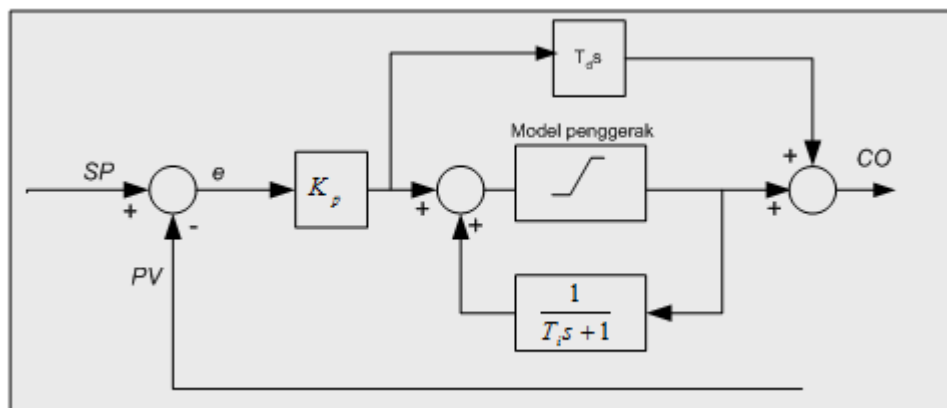
Dalam subbab ini akan dibahas dua buah fitur penting yang umumnya dijumpai pada modul kontrol PID komersil:

- (1) AntiWindup : kemampuan kontroler untuk mengatasi fenomena yang dikenal dengan istilah WindUp
- (2) BumpLess Transfer: kemampuan kontroler untuk beralih mode kontrol tanpa menimbulkan hentakan sinyal kontrol.

### 2.4.1 Phenomena *WindUp* pada kontrol PID dan Antisipasinya

*WindUp* adalah sebuah fenomena yang disebabkan oleh terjadinya saturasi pada penggerak. Saturasi dapat terjadi karena beban yang dikontrol sudah diluar kemampuan penggerak, Jika kontroler tidak memiliki AntiWindup, maka dengan terjadinya beban yang berlebihan, output integrator pada kontroler PID akan terus menerus membesar (*WindUp*), sedangkan disisi lain pembesaran output tersebut tidak dapat lagi mengubah posisi penggerak (karena telah saturasi). Jika misal pada satu saat beban yang dikontrol kembali menjadi normal, maka output kontroler akan memerlukan waktu yang relative lama untuk menuju nilai nominalnya kembali.

Untuk menghindari fenomena tersebut maka dalam kontroler-kontroler komersil umumnya telah dipasang AntiWindup. Gambar 2.49 dibawah memperlihatkan salah satu struktur PID yang telah dilengkapi AntiWindup.:



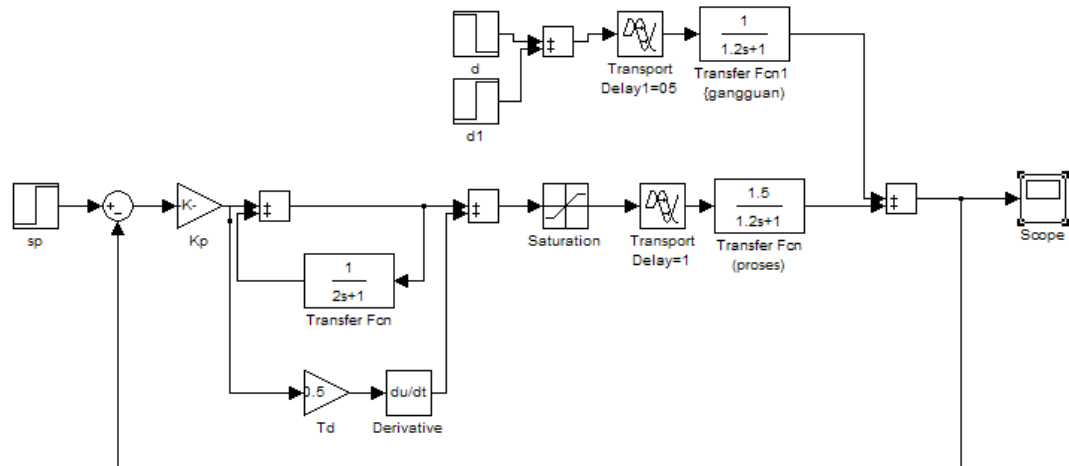
**Gambar 2.49.** Diagram blok kontrol PID dengan Anti Windup

Seperti terlihat pada gambar 2.49 diatas, AntiWindup ini pada dasarnya adalah sebuah model (saturasi) penggerak yang dipasang pada sisi umpan maju kontroler (tentang model saturasi, penjelasannya dapat dilihat pada kotak **catatan penting tentang model saturasi**).

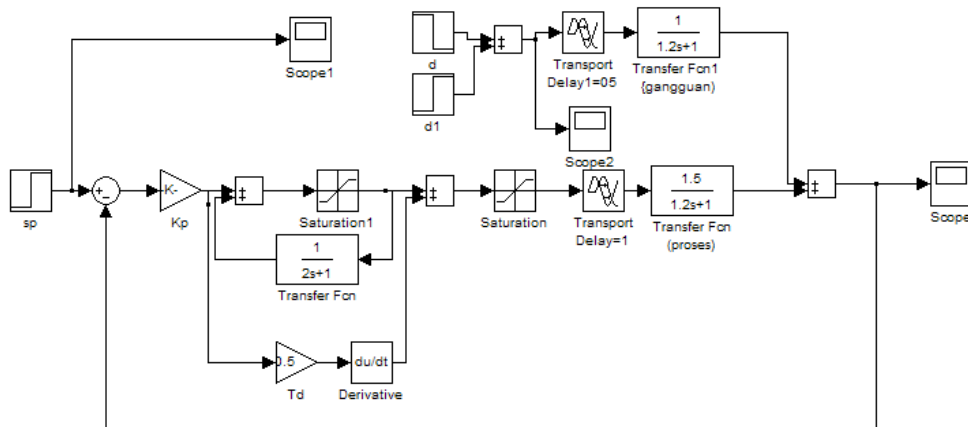
Batas saturasi model penggerak biasanya diseting pada batas saturasi dari penggerak aslinya. Tetapi walaupun demikian, untuk kepentingan keamanan operasi pengontrolan, batas tersebut dapat juga diseting pada batas yang lebih kecil dari nilai saturasi penggeraknya.

Untuk memahami fenomena *Windup* ini akan sangat membantu jika kita gunakan perangkat bantu Simulink. Gambar 2.50 berikut memperlihatkan blok diagram simulink

kontrol PID tanpa kompensasi AntiWindUp dan dengan kompensasi atau komponen AntiwindUp (dalam kedua gambar tersebut penggerak proses dimodelkan secara eksplisit sebagai sebuah blok saturasi).



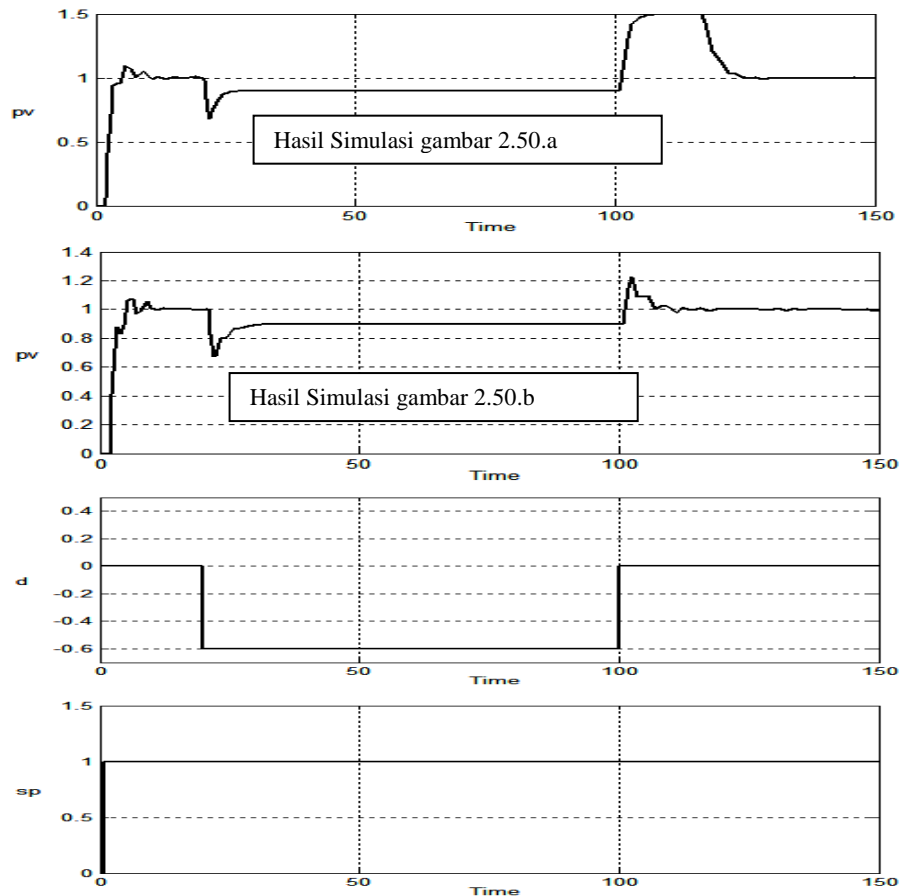
a. Diagram blok simulink dengan sebuah model saturasi penggerak



b. Diagram blok simulink dengan sebuah model saturasi penggerak dan komponen AntiWindUp

**Gambar 2.50.** Blok diagram Simulink pengontrolan PID untuk sebuah proses dengan model saturasi penggerak

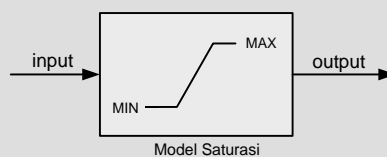
Dengan kondisi pembebanan yang sama, hasil simulasi kedua diagram blok tersebut dapat dilihat pada gambar 2.51 dibawah



**Gambar 2.51.** Salah satu hasil simulasi untuk blok diagram simulink pada gambar 2.50

#### Catatan penting tentang model saturasi

Model saturasi adalah sebuah model non linear (lihat gambar 2.52 dibawah).



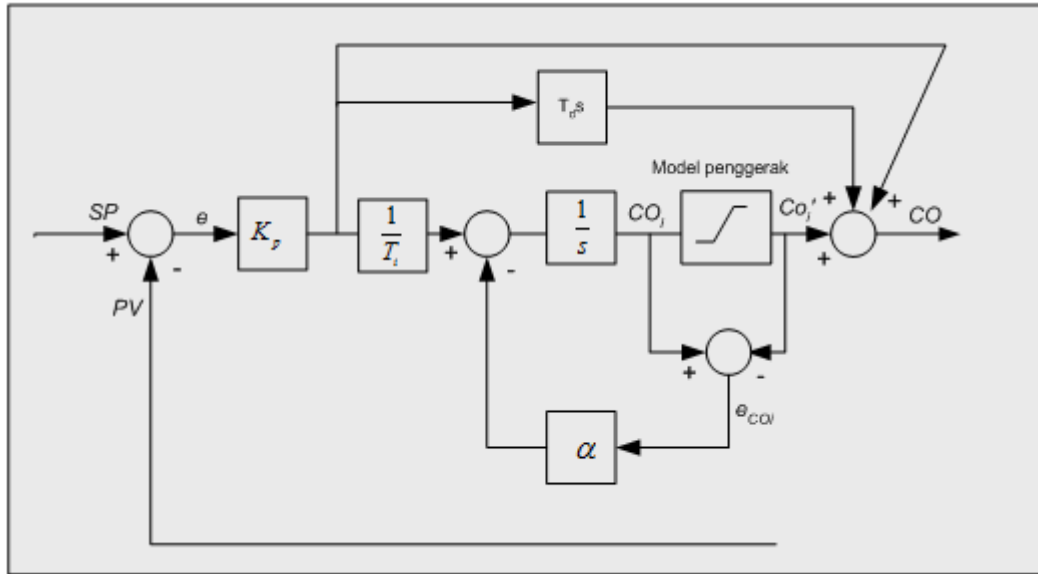
**Gambar 2.52.** Diagram blok model saturasi

Hubungan input –output model ini secara logika dapat ditulis:

If (input ≤ MIN) output = MIN  
If (input ≥ MAX) output = MAX  
Else output = input

Model saturasi pada dasarnya dapat saja menggambarkan hubungan input-output komponen *control valve*: dalam hal ini batas saturasi MIN artinya valve tertutup penuh, sedangkan batas saturasi MAX artinya kondisi valve terbuka penuh. Jika sinyal kontrol dari output controller nilainya memiliki jangkauan ternormalisasi antara 0 - 100(%). Maka valve akan berada dalam batas saturasi MIN jika output kontrol bernilai 0(%) atau kurang, dan akan berada dalam batas saturasi MAX jika output kontrol bernilai 100(%) atau lebih.

Selain gambar 2.49. AntiwindUp pada kontroler PID dapat juga direalisasikan seperti pada gambar 2.53 berikut:



**Gambar 2.53.** Diagram blok alternative kontrol PID (ideal) dengan Anti Windup

Dengan mengacu pada gambar 2.53 diatas, output kontroler dapat dirumuskan:

$$CO(s) = K_p e(s) + CO_i'(s) + K_p T_d s e(s) \quad (2.36)$$

Dalam hal ini  $CO_i'(s)$  adalah output aktual dari blok kontrol integrator:

$$CO_i'(s) = \frac{1}{s} \left[ \frac{K_P}{T_i} e(s) - \alpha (CO_i(s) - CO_i'(s)) \right] \quad (2.37)$$

Berdasarkan persamaan 2.37 tersebut, nampak bahwa jika output integrator hasil perhitungan ( $CO_i$ ) belum mengalami saturasi maka persamaan output integrator aktual akan direduksi menjadi output integrator normal :

$$CO_i'(s) = \frac{K_p}{T_i s} [e(s)] \quad (2.38)$$

Sedangkan jika terjadi saturasi ( $CO_i \geq \text{MAX}$  atau  $CO_i \leq \text{MIN}$ ) maka output integrator aktual akan direset oleh suku  $\alpha(CO_i(s) - CO_i'(s))$  seperti tertulis pada persamaan 2.37 (dalam hal ini besar faktor pe-reset umumnya berada dalam jangkauan :  $1T_i \leq \alpha \leq 10T_i$ ).

### 2.4.2 BumpLess Transfer

Seperti yang telah sekilas dibahas pada **kotak Catatan Penting tentang Mode Kontrol pada Modul PID komersil** (lihat **Bab 1**). Modul PID komersil umumnya menyediakan minimal dua mode pengontrolan yang dapat digunakan secara bebas oleh operator: Mode Manual dan Mode Auto.

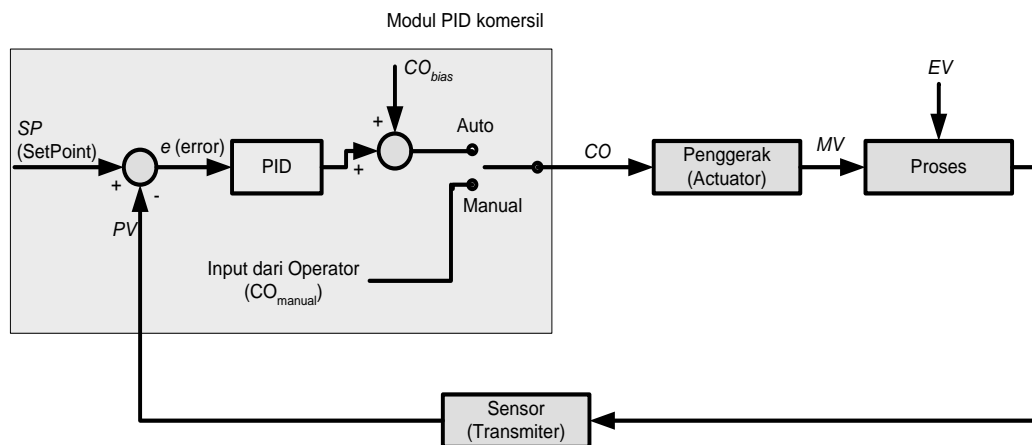
Pengontrolan mode Manual adalah pengontrolan dengan menggunakan prinsip *open loop* dimana besarnya output Kontroler –  $CO$  diatur oleh operator, sedangkan Pengontrolan mode Auto (PID) adalah pengontrolan dengan menggunakan prinsip *closed loop* dimana kontroler bekerja berdasarkan error.

Tergantung kondisi operasi, perubahan mode kontrol dari Manual ke Auto atau sebaliknya adalah hal yang lumrah dilakukan oleh operator.

Salah satu hal penting yang perlu diperhatikan pada saat peralihan mode adalah perbedaan nilai sinyal kontrol awal antara output kontrol PID (Auto) dan output kontrol manual. Jika satu saat misalnya operator mengubah mode dari Manual ke Auto maka bisa saja terjadi hentakan sinyal kontrol yang disebabkan oleh perbedaan sinyal kontrol awal tersebut.

Untuk menghindari hal diatas, maka pada saat peralihan mode, output kontrol PID dalam modul komersil umumnya akan diinisialisasi dengan nilai sinyal kontrol Manual yang sebelumnya telah diseting oleh operator, sedangkan nilai Setpoint – $SP$  biasanya diinisialisasi dengan nilai output proses— $PV$  yang terbaca pada saat itu (Dalam buku manual peralatan hal tersebut dikenal dengan istilah *BumpLess Transfer*).

Dalam blok kontrol PID, nilai inisialisai awal tersebut secara teknis akan disimpan sebagai kontrol bias ( $CO_{bias}$ ) seperti nampak pada gambar 2.54.



**Gambar 2.54.** Sinyal kontrol bias pada modul PID

Dengan mengacu gambar 2.54 diatas, maka output kontrol PID pada dasarnya dapat ditulis secara lengkap dengan mencantumkan sinyal kontrol bias secara eksplisit. Sebagai contoh, jika struktur PID yang digunakan adalah struktur ideal maka output kontrol PID selanjutnya dapat ditulis:

$$CO(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] + CO_{bias} \quad (2.39)$$

Perhatikan, jika sinyal kontrol bias pada modul PID dapat dimanipulasi oleh operator pada saat pengontrolan sedang berjalan, maka penggunaan kontrol bias tersebut selain sangat penting pada saat peralihan mode kontrol (yaitu untuk merealisasikan *BumpLess Transfer*), juga bermanfaat menghilangkan offset yang mungkin terjadi pada saat pengontrolan sedang berjalan (kasus ini terjadi misal jika proses *self regulating* dikontrol dengan menggunakan komponen Proporsional saja)

---

Walaupun berperan dalam proses pengontrolan, besar sinyal kontrol bias ini secara fungsional pada dasarnya tidak akan mempengaruhi kestabilan sistem kontrol umpan balik. Sehingga dengan demikian dalam banyak analisis dinamika sistem, sinyal bias ini umumnya tidak dicantumkan secara eksplisit dalam algoritma kontrol PID.

## 2.5. Contoh Soal dan Penyelesaian

### Contoh 2.1.

#### Tuning PID untuk model Kontrol Proses FOPDT

Diketahui sebuah proses FOPDT dengan fungsi alih :

$$pv(s) = \frac{1.5}{1.2s + 1} e^{-s} .co(s) + \frac{1}{1.2s + 1} e^{-0.5s} .d(s)$$

Jika proses tersebut akan dikontrol menggunakan PID struktur ideal, carilah parameter kontrolnya berdasarkan metode-metode tuning berikut:

- Metode Ziegler Nichols
- Metode Astrom-Haglung

### Penyelesaian

Untuk mendapatkan konstanta-konstanta kontroler PID tersebut, model proses yang perlu diketahui untuk perhitungan hanyalah model proses yang merelasikan antara perubahan sinyal kontrol- $co$  terhadap perubahan output proses- $pv$  (Model gangguan atau pengaruh gangguan - $d$  terhadap perubahan output proses ( $pv$ ) pada dasarnya tidak relevan dalam perhitungan konstanta PID, tetapi penting untuk keperluan simulasi):

$$pv(s) = \frac{1.5}{1.2s + 1} e^{-s} .co(s)$$

dengan  $K = 1.5$

$$T = 1.2$$

$$L = 1$$

Berdasarkan perumusan pada tabel 2.7, maka besarnya konstanta-konstanta kontroler dapat dihitung:

- Metode Ziegler-Nichols:

$$K_p = \frac{1.2T}{KL} = \frac{1.2 \times 1.2}{1.5 \times 1} = 0.96$$

$$T_i = 2L = 2 \times 1 = 2$$

$$T_d = 0.5L = 0.5 \times 1 = 0.5$$

- Metode Astrom-Haglung



$$K_p = \frac{0.94T}{KL} = \frac{0.94 \times 1.2}{1.5 \times 1} = 0.75$$

$$T_i = 2L = 2 \times 1 = 2$$

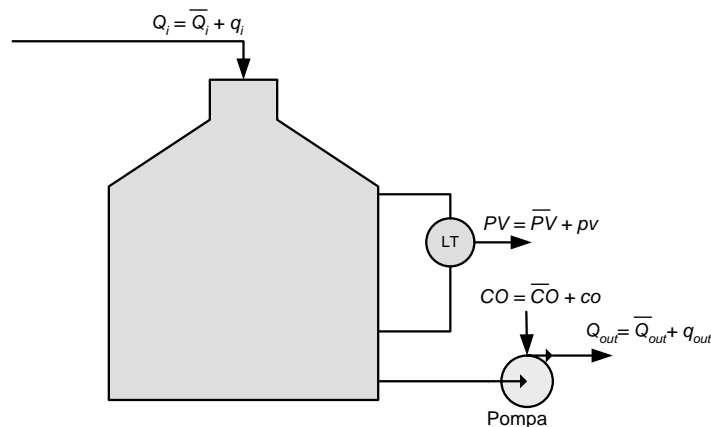
$$T_d = 0.5L = 0.5 \times 1 = 0.5$$

### Contoh 2.2.

#### Tuning PID untuk model Kontrol Proses IPDT

Berikut ini adalah persamaan dinamika sistem pengontrolan level fluida dalam tangki dengan menggunakan penggerak pompa disekitar titik kerjanya (lihat gambar 2.55)

$$H(s) = \frac{pv(s)}{co(s)} = \frac{-0.015e^{-0.1s}}{s}$$



**Gambar 2.55.** Sistem kontrol level fluida dengan penggerak pompa

Jika proses tersebut akan dikontrol menggunakan Kontrol PI,

- Carilah besar parameter yang harus dituning berdasarkan metode metode Shinsky
- Tentukan mode kontroler yang harus digunakan untuk proses tersebut serta gambar pula diagram blok kontrolernya.

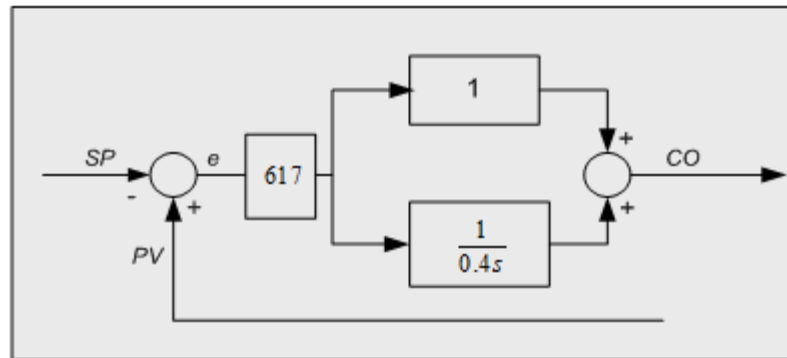
#### Penyelesaian

- Dengan mengacu tabel 2.5, besar penguat berdasarkan metode Shinsky dapat dihitung:

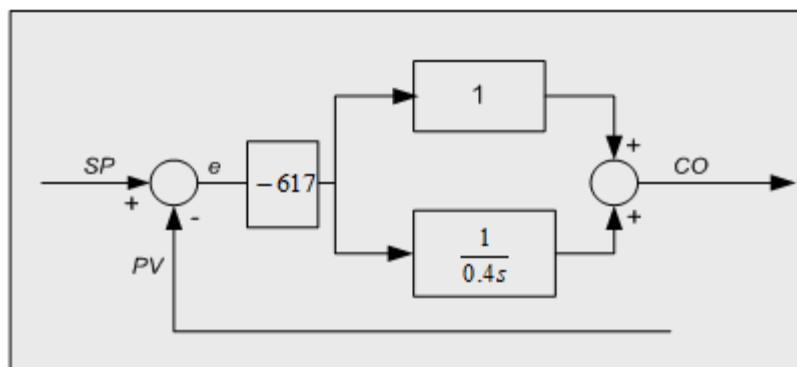
$$K_p = \frac{0.9259}{K^* L} = \frac{0.9259}{0.015 \times 0.1} = 617$$

$$T_i = 4L = 4 \times 0.1 = 0.4$$

- Berdasarkan persoalan, diketahui bahwa proses tersebut pada dasarnya bersifat *Riverse* (tanda gain integratif bernilai negative), sehingga dengan demikian mode kontroler yang harus digunakan adalah mode *Direct*, dimana secara fungsional mode ini dapat diilustrasikan seperti nampak pada gambar 2.56 atau gambar 2.57.



**Gambar 2.56.** Diagram blok solusi kontrol PI (mode kontrol Reverse)



**Gambar 2.57.** Diagram blok alternatif solusi kontrol PI (mode kontrol Reverse)

### Contoh 2.3.

#### Blok kontrol PI

Buktikan bahwa secara matematis diagram blok dalam gambar 2.13 akan sama dengan persamaan kontrol PI dalam persamaan 2.10

#### Penyelesaian

Berdasarkan gambar 2.13 terlihat bahwa

$$CO(s) = K_p e(s) + \frac{1}{T_i s + 1} CO(s)$$

atau dapat ditulis:

$$\frac{T_i s}{T_i s + 1} CO(s) = K_p e(s)$$

$$CO(s) = K_p \frac{T_i s + 1}{T_i s} e(s)$$

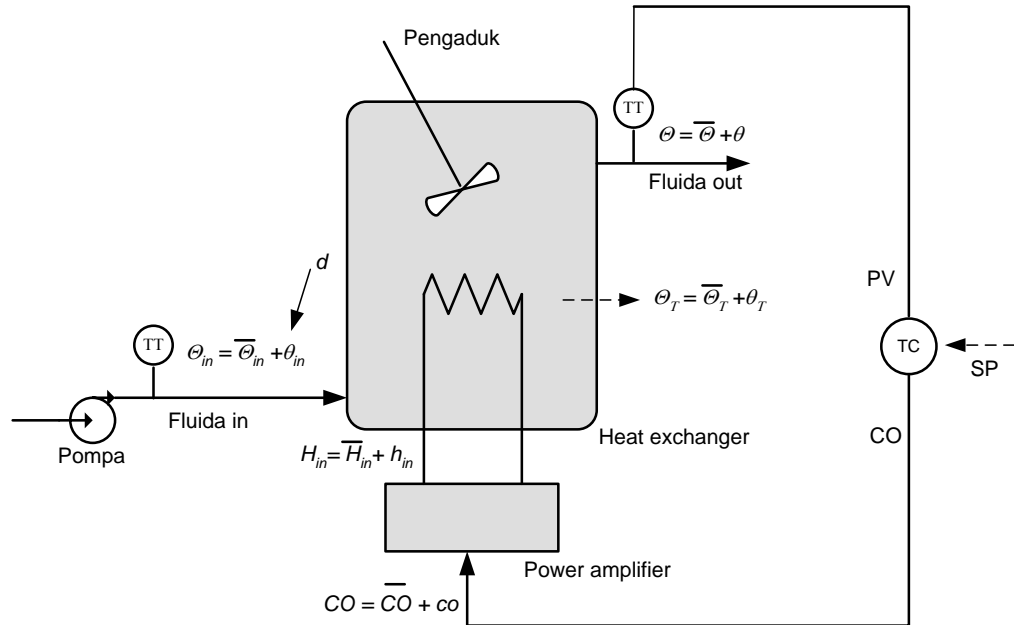
$$CO(s) = K_p e(s) + \frac{K_p}{T_i s} e(s)$$

dengan mengumpulkan suku penguat proporsional, maka didapat:

$$CO(s) = K_p \left[ 1 + \frac{1}{T_i s} \right] e(s)$$

**Contoh 2.4.**  
**Sistem Kontrol Heat Exchanger**

Tinjau sistem kontrol temperature pada Heat Exchanger berikut



**Gambar 2.58.** Sistem kontrol Heat Exchanger

Berdasarkan hasil penyelesaian pada **contoh soal 1.8** (lihat **Bab 1**), hubungan antara perubahan sinyal control dan perubahan temperature fluida inlet terhadap perubahan fluida outlet memiliki dinamika berikut:

$$\theta(s) = \frac{Ke^{-Ls}}{Ts + 1} co(s) + \frac{e^{-Ls}}{Ts + 1} \theta_{in}(s)$$

dimana

$$T = \frac{V \cdot \rho}{w} \text{ (menit)}$$

$$K = \frac{K_{ps}}{cw} \text{ (} ^\circ\text{C/\% )}$$

$$L = \frac{K_L}{w} \text{ (menit)}$$

dalam hal ini

$V$  = Volume tangki

$w$  = Laju aliran fluida dalam keadaan steady

---

$\rho$  = Densitas fluida  
 $c$  = Kalor jenis fluida  
 $K_{ps}$  = konstanta kesebandingan antara perubahan panas dengan perubahan sinyal kontrol  
 $K_L$  = Konstanta keterlambatan (kg/menit) menit

Jika misalnya semua parameter diatas nilainya konstan kecuali laju aliran fluida, yaitu masing-masing bernilai:

$$\begin{aligned}V &= 0.2 \text{ m}^3 \\ \rho &= 1000 \text{ kg/m}^3 \\ c &= 4000 \text{ J/(Kg} \cdot ^\circ\text{C)} \\ K_{ps} &= 48000 \text{ J/menit/}\% \\ K_L &= 50 \text{ (Kg/menit)menit}\end{aligned}$$

Carilah nilai parameter PID (tipe B) berdasarkan metode tuning Van Doren untuk masing-masing untuk laju aliran fluida berikut:

- 20 kg/menit
- 50 kg/menit
- 80 kg/menit

### Penyelesaian

Berdasarkan tabel 2.9 diketahui perumusan tuning Van doren:

$$K_p = \frac{1.5T}{K.L}$$

$$T_i = 2.5L$$

$$T_d = 0.4L$$

Dalam hal ini:

$$T = \frac{V\rho}{w} = \frac{0.2 \times 1000}{w} \text{ (menit)}$$

$$K = \frac{K_{ps}}{cw} = \frac{48000}{4000 \times w} \text{ (}^\circ\text{C/}\% \text{)}$$

$$L = \frac{K_L}{w} = \frac{50}{w} \text{ (menit)}$$

- Untuk laju aliran fluida ( $w$ ) = 20 kg/menit, didapat:

$$T = \frac{V\rho}{w} = \frac{0.2 \times 1000}{20} = 10 \text{ (menit)}$$

$$K = \frac{K_{ps}}{cw} = \frac{48000}{4000 \times 20} = 0.6 \text{ (}^\circ\text{C/}\% \text{)}$$

$$L = \frac{K_L}{w} = \frac{50}{20} = 2.5 \text{ (menit)}$$

Sehingga dengan demikian besar parameter kontrolernya berturut-turut :

$$K_p = \frac{1.5T}{K.L} = \frac{1.5 \times 10}{0.6 \times 2.5} = 10$$

$$T_i = 2.5L = 2.5 \times 2.5 = 6.25$$

$$T_d = 0.4L = 0.4 \times 2.5 = 1$$

- Untuk laju aliran fluida ( $w$ ) = 50 kg/menit, didapat:

$$T = \frac{V\rho}{w} = \frac{0.2 \times 1000}{50} = 4 \text{ (menit)}$$

$$K = \frac{K_{ps}}{cw} = \frac{48000}{4000 \times 50} = 0.24 \text{ (}^\circ\text{C/\% )}$$

$$L = \frac{K_L}{w} = \frac{50}{50} = 1 \text{ (menit)}$$

Sehingga dengan demikian besar parameter kontrolernya berturut-turut :

$$K_p = \frac{1.5T}{K.L} = \frac{1.5 \times 4}{0.24 \times 1} = 25$$

$$T_i = 2.5L = 2.5 \times 1 = 2.5$$

$$T_d = 0.4L = 0.4 \times 1 = 0.4$$

- Untuk laju aliran fluida ( $w$ ) = 80 kg/menit, didapat:

$$T = \frac{V\rho}{w} = \frac{0.2 \times 1000}{80} = 2.5 \text{ (menit)}$$

$$K = \frac{K_{ps}}{cw} = \frac{48000}{4000 \times 80} = 0.15 \text{ (}^\circ\text{C/\% )}$$

$$L = \frac{K_L}{w} = \frac{50}{80} = 0.625 \text{ (menit)}$$

Sehingga dengan demikian besar parameter kontrolernya berturut-turut :

$$K_p = \frac{1.5T}{K.L} = \frac{1.5 \times 2.5}{0.15 \times 0.625} = 40$$

$$T_i = 2.5L = 2.5 \times 0.625 = 1.56$$

$$T_d = 0.4L = 0.4 \times 0.625 = 0.25$$

---

### Contoh 2.5

#### Perhitungan parameter PID serial VS paralel

Untuk sebuah proses tertentu, diketahui bahwa PID serial memiliki besar parameter optimal berikut:

$$K_p = 0.5$$

$$T_i = 2$$

$$T_d = 1.8$$

Jika proses yang sama akan dikontrol dengan menggunakan PID ideal, carilah parameter kontrol optimal yang harus diseting pada kontroler tersebut.

#### Penyelesaian

Dengan menggunakan persamaan 2.26, 2.27 dan 2.28, besar parameter PID idealnya berturut-turut didapat:

$$K_P = K_{PS} \frac{T_{IS} + T_{DS}}{T_{IS}} = 0.5 \frac{2 + 1.8}{2} = 0.95$$

$$T_I = T_{IS} + T_{DS} = 2 + 1.8 = 3.8$$

$$T_D = \frac{T_{IS} T_{DS}}{T_{IS} + T_{DS}} = \frac{2 \times 1.8}{2 + 1.8} = 0.95$$

### Contoh 2.6

#### Perhitungan parameter kontrol PI untuk proses orde satu

Diketahui sebuah proses memiliki dinamika sistem orde satu berikut:

$$H(s) = \frac{pv(s)}{co(s)} = \frac{K}{Ts + 1}$$

dalam hal ini  $K$  = gain statis proses

$T$  = konstanta waktu proses

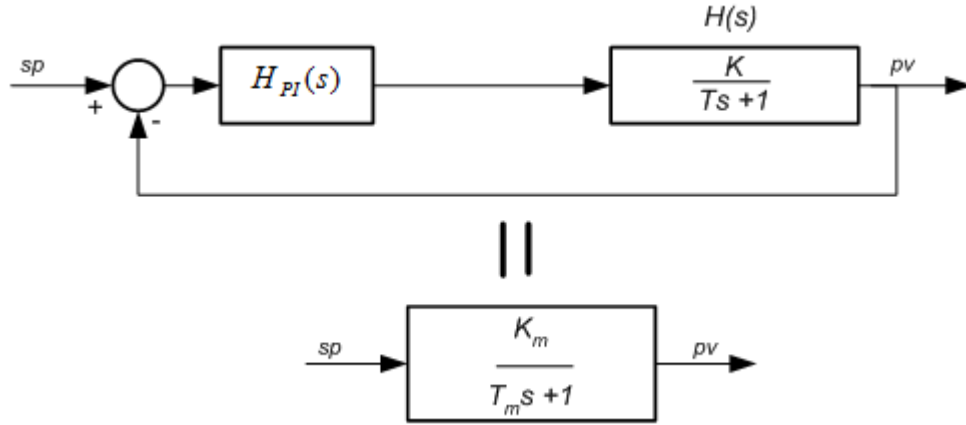
Jika proses diatas akan dikontrol dengan menggunakan kontroler PI, carilah parameter kontroler tersebut sedemikian sehingga dinamika atau fungsi alih antara perubahan Setpoint- $sp$  terhadap perubahan output proses -  $pv$ , memiliki fungsi alih ideal berikut (perhatikan juga ilustrasinya pada gambar 2.63):

$$H_{cl}(s) = \frac{pv(s)}{sp(s)} = \frac{K_m}{T_m s + 1} = \frac{1}{T_m s + 1}$$

dalam hal ini

$K_m$  = gain statis umpan balik proses (agar dalam keadaan steady-nya, output proses = Setpoint, maka nilai  $K_m$  harus dipilih sama dengan satu)

$T_m$  = konstanta waktu umpan balik proses (nilai ditentukan oleh perancang tergantung dari kecepatan respon yang diinginkan)



**Gambar 2.59.** Sistem umpan balik kontrol PI untuk proses orde satu

### Penyelesaian

Diketahui fungsi alih kontrol PI:

$$H_{PI}(s) = K_p \left[ 1 + \frac{1}{T_I s} \right]$$

atau dapat ditulis ulang:

$$H_{PI}(s) = \frac{K_p \left[ s + \frac{1}{T_I} \right]}{s}$$

Berdasarkan gambar 2.59, fungsi alih antara  $pv$  terhadap  $sp$  :

$$H_{cl}(s) = \frac{pv(s)}{sp(s)} = \frac{H_{PI}(s)H(s)}{1 + H_{PI}(s)H(s)}$$

Dengan mensubsitusikan setiap elemen umpan balik  $H_{PI}(s)$  dan  $H(s)$  pada persamaan diatas, serta dengan sedikit langkah penyederhanaan, maka didapat::

$$H_{cl}(s) = \frac{pv(s)}{sp(s)} = \frac{K_p K \left[ s + \frac{1}{T_I} \right]}{s[Ts + 1] + K_p K \left[ s + \frac{1}{T_I} \right]}$$

Agar

$$H_{cl}(s) = \frac{1}{sT_{cl} + 1}$$

Maka

$$s^2T + s = K_p K T_{cl} s^2 + \frac{K_p K T_{cl}}{T_I} s$$

Untuk besar gain statis proses- $K$  dan konstanta waktu proses- $T$  yang diketahui, maka besarnya penguat proporsional  $-K_p$  serta konstanta waktu integral  $-T_I$  dapat diperoleh berdasarkan relasi dibawah ini:

$$K_p = \frac{T}{T_{cl} K}$$

$$T_I = T$$

Dalam hal ini besarnya konstanta waktu umpan balik proses  $T_{cl}$  harus ditentukan terlebih dulu dalam perancangan: jika diinginkan respon proses berjalan cepat maka  $T_{cl}$  yang diberikan harus bernilai kecil (tetapi respon proses sebenarnya tentunya sangat tergantung kekuatan dari penggerak proses tersebut).

### Contoh 2.7

#### Studi kasus Perhitungan parameter kontrol PI untuk proses orde satu

Diketahui sebuah model proses orde 1 memiliki besar gain statis  $K = 0.5$  dan konstanta waktu proses  $T = 0.3$  (menit). Jika proses tersebut akan dikontrol dengan kontroler PI, carilah besar penguat proporsional  $-K_p$  dan konstanta waktu integral  $-T_I$  sedemikian sehingga dinamika perubahan output proses- $pv$  terhadap perubahan Setpoint- $sp$  memiliki konstanta waktu 0.2 (menit)

#### Penyelesaian

Berdasarkan persamaan 2.35 atau hasil penyelesaian contoh soal 2.6, besarnya gain proporsional dan konstanta waktu integral dapat dihitung:

$$K_p = \frac{T}{T_{cl} K} = \frac{0.3}{0.2 \times 0.5} = 3$$

$$T_I = T = 0.3$$

## 2.6 Ringkasan

- Kontroler PID adalah jenis kontroler yang sangat umum digunakan di industri proses. Modul PID komersil pada umumnya dapat dijumpai dalam bentuk modul *Special Process Controller* (misal temperature controller, pressure controller dan sebagainya), modul DCS atau modul PLC.
- Berdasarkan kenyataan, ada beragam jenis struktur kontrol berbasis komponen PID, masing-masing dikenal dengan nama : Struktur PID tipe A, PID tipe B, PID tipe C, PID standar ISA, PID general dan PID serial. Masing-masing struktur umumnya juga dilengkapi filter orde satu pada suku derivatifnya. Hal tersebut dilakukan untuk menekan derau yang pasti muncul dalam pengukuran.
- Untuk setiap struktur PID telah tersedia metode-metode tuning empiris yang dapat digunakan untuk menentukan besarnya parameter kontrol optimal untuk sebuah model proses yang dihadapi.
- Secara umum, setiap komponen kontrol PID memiliki fungsi sebagai berikut:



- Kontrol Proporsional : berfungsi untuk mempercepat respon
- Kontrol Integral : berfungsi untuk menghilangkan error steady
- Kontrol Derivative : berfungsi untuk memperbaiki respon transien.
- Dalam bentuk simulasi lengkapnya, semua variabel input dan output pada setiap blok simulasi (yaitu blok kontroler serta blok proses) harus diinterpretasikan dan dipahami sebagai variabel deviasi disekitar nilai nominalnya.
- Bumpless transfer adalah salah satu fitur pada modul kontrol komersil yang berfungsi untuk menghindari perubahan output kontroler yang relative besar ketika terjadi perubahan mode kontroler dari mode Auto ke mode manual atau sebaliknya
- Tidak tergantung metode tuning yang digunakan, struktur kontrol PID pada dasarnya akan menghasilkan unjukkerja yang kurang jika proses yang dikontrol memperlihatkan delay transportasi yang dominan ( $L/T > 1$ )
- Untuk proses-proses yang memiliki delay transportasi dominan, maka struktur kontrol PID yang digunakan harus memiliki kompensator tambahan. Dua buah struktur kontrol yang dapat digunakan untuk mengontrol proses tersebut adalah Smith Predictor dan predictive PI (pPI)

## 2.7 Soal Latihan

2.1. Tinjau proses berikut ini:

$$pv(s) = \frac{0.5}{2s+1} e^{-1.5s} .co(s) + \frac{1}{2s+1} e^{-0.5s} d(s)$$

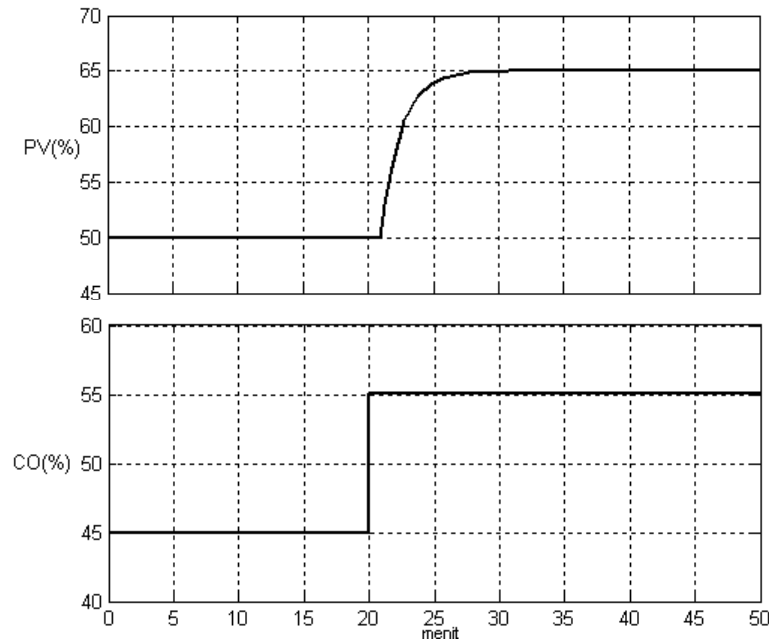
Pertanyaan.

- a. Dengan menggunakan metode Ziegler-Nichols, Carilah besar parameter kontrol Proporsional untuk proses tersebut
  - b. Dengan menggunakan metode Ziegler-Nichols, Carilah besar parameter kontrol Proporsional Integral (PI) untuk proses tersebut
  - c. Dengan menggunakan metode Ziegler-Nichols, Carilah besar parameter kontrol Proporsional Integral (PID) ideal untuk proses tersebut
  - d. Untuk masing-masing jawaban a-c simulasikan dan analisis hasilnya dengan menggunakan perangkat lunak Simulink
- 2.2. Dengan menggunakan Simulink, buktikan bahwa penggunaan metode kontrol Proporsional pada model proses IPDT akan selalu menghasilkan error steady jika pada proses tersebut terdapat gangguan.

Sebagai studi kasus gunakan model proses berikut:

$$pv(s) = \frac{-0.5}{s} e^{-1.2s} .co(s) + \frac{0.1}{s} d(s)$$

2.3. Tinjau grafik relasi hubungan perubahan sinyal kontrol vs perubahan output proses pada gambar 2.60 berikut.



**Gambar 2.60.** Grafik relasi CO vs PV untuk soal 2.3

Untuk proses tersebut, carilah konstanta-konstanta PID optimal berdasarkan metode yang tersedia dalam buku ini.

2.4. Rancanglah kontroler Smith Predictor untuk proses :

$$pv(s) = \frac{0.5}{2s+1} e^{-1.5s} .co(s) + \frac{1}{2s+1} e^{-0.5s} d(s)$$

2.5 Untuk sebuah proses tertentu, diketahui bahwa PID paralel memiliki besar parameter optimal berikut:

$$K_p = 0.5$$

$$T_i = 2$$

$$T_d = 1.8$$

Jika proses yang sama akan dikontrol dengan menggunakan PID serial, carilah parameter kontrol optimal yang harus diseting pada kontroler tersebut.

2.6 Buktikan bahwa diagram blok Smith Predictor gambar 2.43 pada dasarnya ekuivalen dengan diagram blok gambar 2.44

ooOoo

## Bab 3

# AUTOTUNING DAN GAIN SCHEDULING

### Tujuan Bab

Ada dua materi penting yang akan dibahas secara mendalam dalam bab ini: (1) Autotuning dan (2) Gain Scheduling. Kedua kemampuan tersebut umum dijumpai pada modul kontrol PID komersil. Setelah menyelesaikan bab ini, anda diharapkan dapat:

- Menjelaskan prinsip kerja Autotuning
- Menjelaskan Metode Autotuning Respon Transien
- Menjelaskan prinsip kerja sistem kontrol umpan balik relay
- Menjelaskan Metode Autotuning Umpan Balik relay (*relay Feedback*)
- Menyebutkan kelebihan dan kelemahan masing-masing metode Autotuning
- Menjelaskan prinsip kerja Gain Scheduling
- Menyebutkan variabel-variabel penjadwal (*scheduling*) yang umum digunakan pada metode Gain Scheduling

### Bacaan Prasyarat

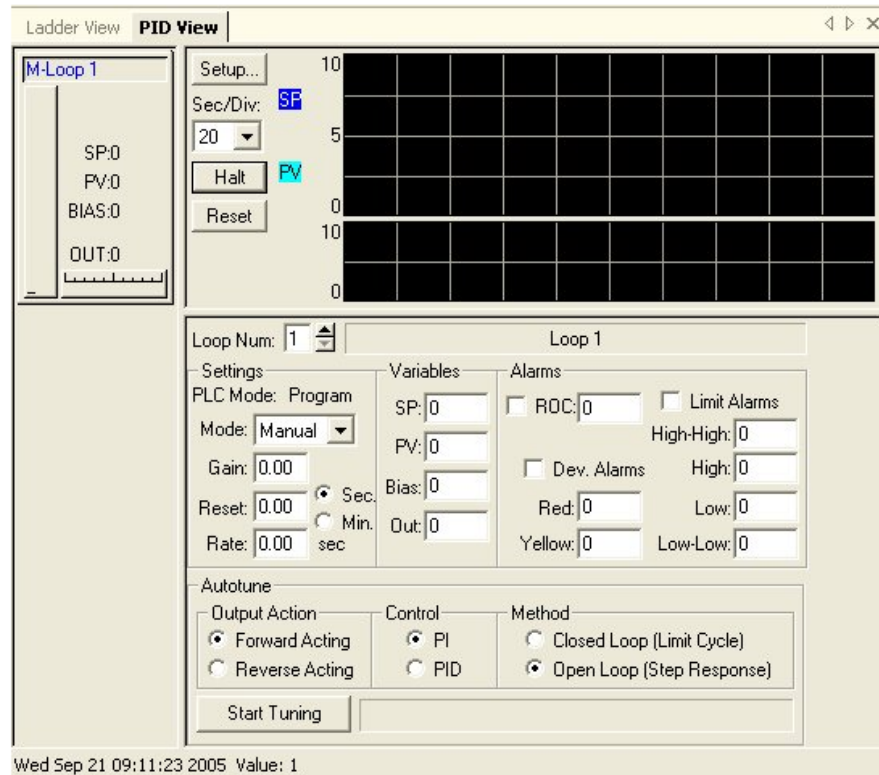
- Bab 1. Dinamika dan Model Kontrol Proses Industri
- Bab 2. Kontrol PID Untuk Proses Industri : Berbagai Struktur dan Metoda Tuning Praktis

### 3.1 Pendahuluan

Sejalan dengan perkembangan teknologi perangkat keras dan perangkat lunak yang begitu luar biasa, dewasa ini sejumlah besar modul PID yang beredar di pasaran umumnya telah dilengkapi dengan beberapa kemampuan pendukung tambahan diluar fungsi dasar yang telah menjadi standar sebuah kontrol PID komersil. Salah satu diantara kemampuan penting yang dapat dijumpai adalah Autotuning. Dalam beberapa buku manual, Autotuning ini dikenal juga dengan nama *Self-Tuning* atau *Pre-Tuning*, yaitu kemampuan modul untuk menseting parameter PID secara otomatis.

Untuk modul PID ukuran kecil atau modul PID jenis *special process controller*, fasilitas autotuning diaktifkan umumnya lewat penekanan salah satu tombol yang memang disediakan pada modul, sedangkan untuk modul PID ukuran besar atau modul PID pada sistem PLC, fasilitas autotuning tersebut diaktifkan lewat *console* atau perangkat lunak yang berjalan di komputer. Gambar 3.1 memperlihatkan contoh GUI (*Grafical User Interface*) pada perangkat lunak DirectSoft yang menyediakan fasilitas pengaktifan autotuning untuk PLC merk DirectLogic.

Terkait dengan Autotuning, saat ini banyak dijumpai juga perangkat lunak komputer independen yang dikhususkan untuk men-tuning parameter kontroler untuk beragam modul PID dari vendor yang berbeda. Salah satu kelebihan yang ditawarkan oleh perangkat lunak tersebut adalah penggunaan berbagai metode tuning optimal yang disesuaikan dengan karakteristik proses yang teridentifikasi.



**Gambar 3.1.** Fasilitas GUI untuk mengaktifkan fasilitas Autotuning pada perangkat lunak DirectSoft.

Ditinjau dari metode Autotuning yang digunakan, ada dua buah metode yang umum dijumpai pada modul kontrol PID: (1) Metode respon transien, dan (2) Metode umpan balik relay. Apapun metodenya, proses Autotuning pada dasarnya harus dilakukan saat output proses (PV) telah mengalami keadaan *steady* disekitar nilai terharapnya (yaitu di daerah titik kerja proses).

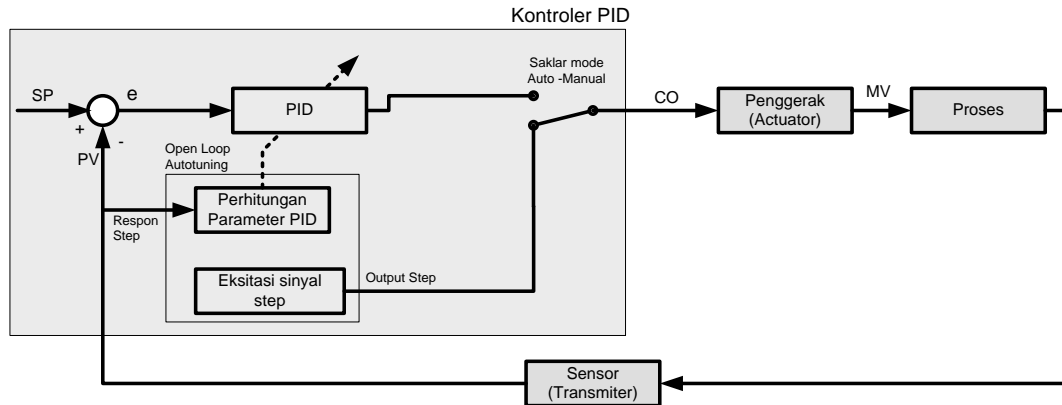
Selain kemampuan Autotuning, beberapa modul PID komersil juga telah dilengkapi fitur Gain Scheduling. Sesuai dengan namanya, metode Gain Scheduling ini memberikan kemampuan pada modul untuk menjadwalkan besaran parameter kontrolnya. Kemampuan tersebut sangat bermanfaat terutama jika proses yang dikontrol memiliki beberapa titik kerja atau kondisi operasi yang berbeda (misal pada proses non linear dan/atau proses yang sering mengalami perubahan beban yang relative ekstrim).

### 3.2 Autotuning Metode Respon Transien

Metode Autotuning Respon transien adalah metode tuning yang berbasiskan eksperimen Bump test (Hanya saja dalam metode Autotuning ini, identifikasi parameter proses beserta kalkulasi paramater kontrolnya dilakukan secara otomatis oleh modul kontroler atau komputer).

Terkait dengan model proses yang dikontrol, secara praktis metode Autotuning ini hanya dapat digunakan untuk proses-proses yang bersifat *self regulating* saja (misal model proses FOPDT). Hal tersebut disebabkan mode kontroler yang digunakan pada saat

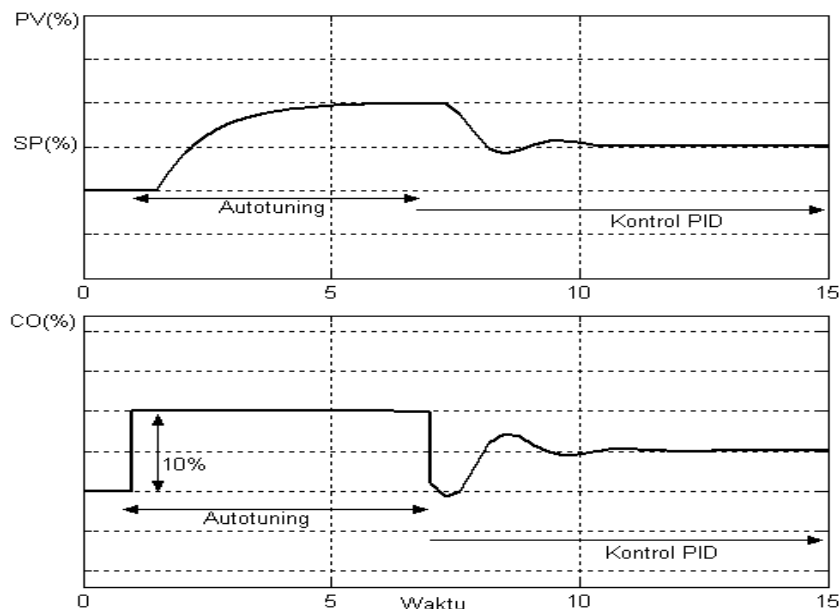
Autotuning berjalan adalah mode manual atau *open loop* (lihat ilustrasinya pada gambar 3.2).



**Gambar 3.2** Ilustrasi metode autotuning respon transien

Seperti halnya prosedur pelaksanaan eksperimen Bump test, langkah awal yang harus dilakukan untuk menjalankan Autotuning ini adalah membawa output proses --PV ke daerah sekitar titik kerja yang diharapkan (Setpoint--SP) terlebih dulu, hal ini dapat dilaksanakan dengan cara mengatur besar sinyal kontrol CO secara manual oleh operator. Setelah PV mengalami *steady* (menetap) di daerah nominalnya, maka proses autotuning tersebut dapat dimulai.

Tergantung modul kontrol PID yang digunakan, nilai perubahan sinyal kontrol (sinyal *step*) yang diinduksikan pada saat autotuning dijalankan umumnya berkisar antara 5%-10%. Untuk banyak kasus, perubahan nilai tersebut pada dasarnya sudah dapat menyebabkan respon proses mengalami deviasi output yang dapat terukur.



**Gambar 3.3** Hubungan sinyal kontrol (CO) vs variabel proses (PV) setelah fasilitas autotuning dijalankan.

Walaupun secara teoritis, besar nilai SetPoint (SP) tidak mempengaruhi jalannya proses Autotuning (karena Autotuning berjalan pada mode Manual), tetapi secara teknis nilai SP

---

dalam modul kontrol umumnya terlebih dulu harus disetting pada kisaran yang tidak jauh dari nilai  $PV$  saat Autotuning akan dimulai. Hal ini disebabkan setelah proses Autotuning selesai, mode kontrol secara otomatis akan berubah menjadi mode Auto dengan nilai setpoint yang telah diseting tersebut. Gambar 3.3 memperlihatkan ilustrasi hubungan sinyal kontrol dengan variabel proses setelah Autotuning dijalankan.

Seperti nampak pada gambar 3.3, sesaat setelah Autotuning diaktifkan, maka secara otomatis kontroler akan menginduksikan perubahan sinyal output kontroler (dalam gambar tersebut besarnya 10% dari nilai output kontrol *steady* awalnya).

Dengan menganggap proses yang dikontrol memiliki model FOPDT jenis *direct*, maka perubahan output kontrol ini akan menyebabkan output proses mengalami deviasi positif dari nilai sebelumnya. Secara teknis perubahan output proses akan selalu dimonitor oleh kontroler: Jika satu saat tidak terjadi lagi perubahan output proses (artinya output proses telah mencapai keadaan *steady* yang baru), maka langkah selanjutnya modul kontrol akan menghitung/mencari parameter proses yang teridentifikasi ( $L, T$  dan  $K$ ). Nilai parameter proses yang teridentifikasi tersebut akan dijadikan parameter masukan pada perumusan metode tuning yang digunakan pada modul.

Tergantung modul kontrol atau perangkat lunak Autotuning-nya, perumusan atau metode tuning yang digunakan bisa saja berasal dari perumusan Ziegler-Nichols, metode Chien, Astrom dan sebagainya (lihat kembali **Bab 2**).

### 3.3 Autotuning Metode Umpan Balik Relay

Autotuning metode umpan balik relay dikenal juga dengan nama Autotuning metode respon frekuensi atau metode siklus terbatas (*limit cycle*). Tidak seperti autotuning metode respon transien, Autotuning metode umpan balik relay secara praktis dapat dilakukan baik untuk proses yang bersifat *self regulating* maupun proses *non self regulating*.

Prinsip kerja metode Autotuning ini pada dasarnya bertumpu pada informasi perioda dan besar magnitude osilasi output proses yang pasti terjadi pada titik kerjanya saat proses dikontrol dengan menggunakan metode kontrol umpan balik relay.

#### 3.3.1. Konsep Dasar Kontrol umpan balik Relay

Kontrol umpan balik relay atau kontrol *ON-OFF* adalah sebuah metode kontrol dimana output kontroler hanya memiliki dua kondisi saja yaitu maximum atau minimum. Berdasarkan prinsip kerjanya, penggunaan kontrol relay ini secara praktis ditujukan agar variabel proses yang dikontrol secepat mungkin menuju level operasi yang diharapkan yaitu dengan cara menginduksikan sinyal kontrol maksimum atau minimum tergantung dari selisih antara setpoint ( $SP$ ) dengan variabel yang dikontrol ( $PV$ ).

Jika output proses yang dikontrol tidak dituntut untuk sama persis dengan nilai Setpoint, maka kontrol umpan balik relay ini pada dasarnya sudah cukup memadai digunakan sebagai basis pengontrolan variabel proses, karena selain prinsip kerjanya yang relative sederhana, sistem kontrol ini juga relative "stabil".

Agar output proses yang dikontrol selalu berada disekitar titik kerja (*Setpoint*) yang diharapkan, maka secara praktis pada output relay ini diperlukan tambahan sinyal kontrol bias yang besarnya harus dapat diatur oleh operator (lihat diagram bloknya pada gambar 3.4).

Secara matematis, output kontrol relay dapat direpresentasikan oleh persamaan 3.1 berikut:

$$CO = CO_R + CO_{bias} \quad (3.1)$$

dalam hal ini:

$$CO_R = A \cdot \text{sgn}(e) \quad (3.2)$$

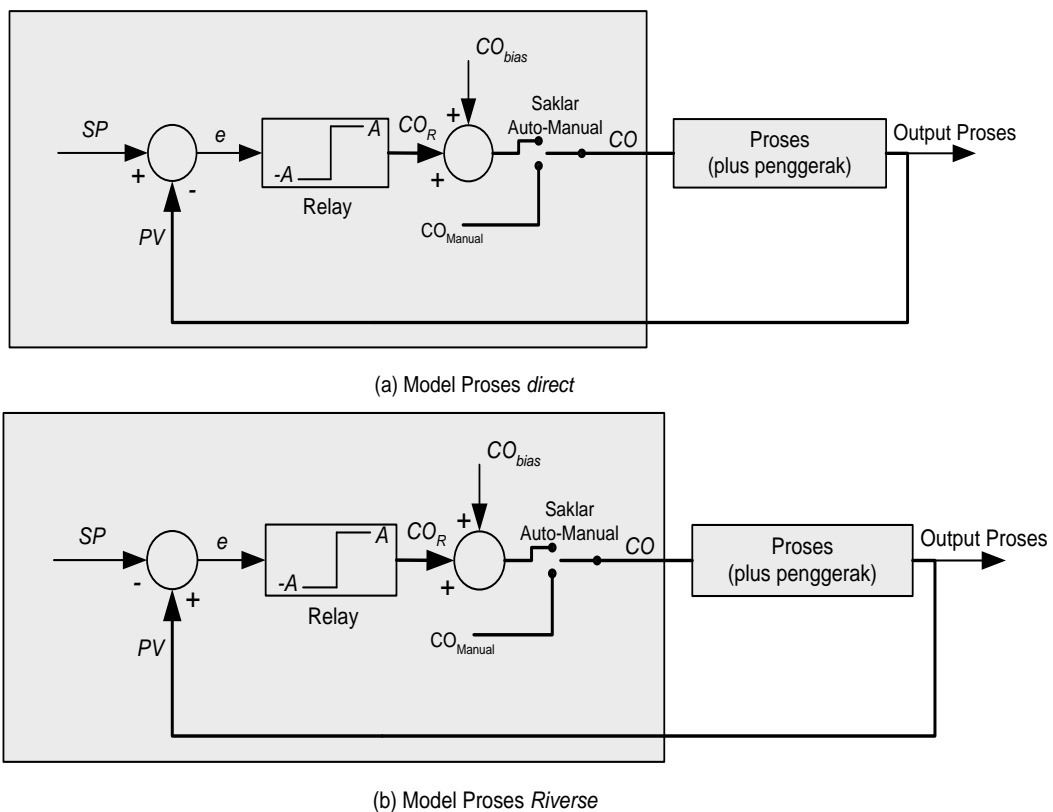
dengan

$A$  = magnitude output relay (umumnya berkisar antara 5%-10% --untuk skala output kontrol : 0% - 100%)

$\text{sgn}(\cdot)$  = fungsi signum (fungsi yang mengembalikan nilai 1 atau -1 tergantung tanda dari sinyal error: jika  $e > 0$ , maka  $\text{sgn}(e) = 1$ , jika  $e < 0$ , maka  $\text{sgn}(e) = -1$ )

$CO_R$  = output relay (memiliki dua kondisi :  $+A$  atau  $-A$ , tergantung dari tanda nilai error—lihat fungsi signum)

$CO_{bias}$  = sinyal bias yang besarnya harus dapat diatur oleh operator sedemikian sehingga output proses berada disekitar daerah kerjanya



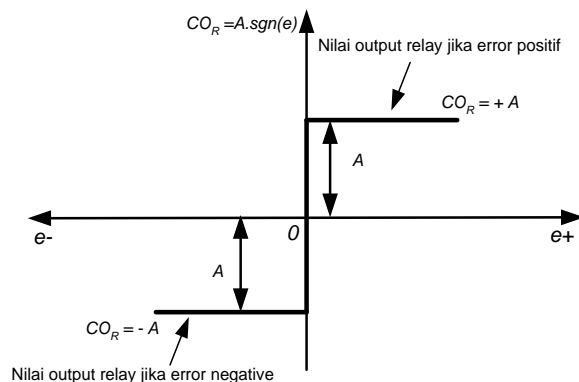
**Gambar 3.4** Diagram blok kontrol proses dengan menggunakan metode kontrol relay

Untuk nilai magnitude  $A$  yang digunakan, relasi input-output pada persamaan (3.1) dan (3.2) tersebut secara fungsional dapat diilustrasikan oleh gambar 3.5.

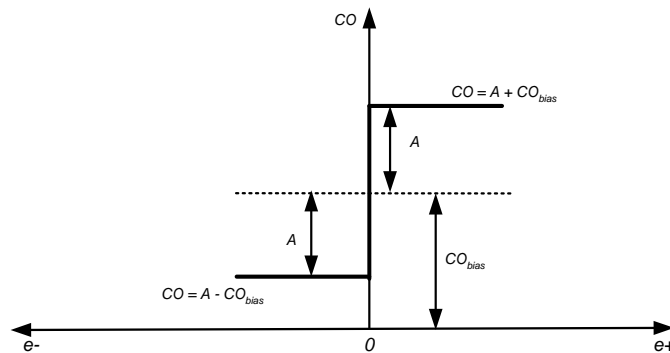
Berdasarkan gambar 3.5(a), nampak bahwa output relay pada dasarnya hanya memiliki dua kondisi saja: Jika error bernilai positive (untuk proses *direct* artinya  $SP > PV$ , sedangkan untuk proses *reverse*  $SP < PV$ ) maka output relay bernilai  $+A$ , dan

sebaliknya jika error bernilai negative (untuk proses *direct* artinya  $SP < PV$ , sedangkan untuk proses *reverse*  $SP > PV$ ) maka output relay bernilai  $-A$ .

Karena output kontroler umumnya memiliki nilai ternormalisasi antara 0% - 100% atau bernilai positive (misal untuk kontrol valve, nilai 0% artinya valve tertutup penuh, sedangkan 100% artinya valve terbuka penuh), dan terutama juga dimaksudkan memberi kebebasan bagi operator untuk membawa proses agar beroperasi disekitar titik kerjanya, maka secara praktis pada output kontrol relay perlu ditambahkan sinyal kontrol bias yang besarnya harus dapat diatur oleh operator sedemikian sehingga proses pengontrolan selalu berada disekitar Setpoint yang diharapkan (lihat gambar 3.5(b)).



(a). Hubungan antara error terhadap output relay



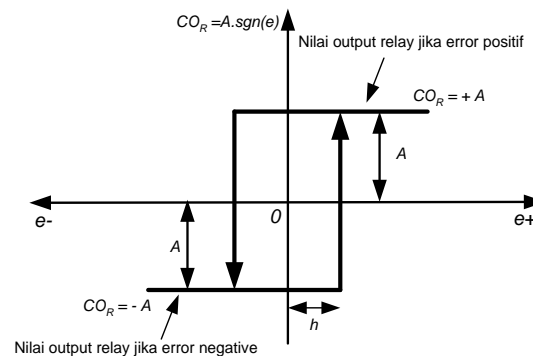
(b). Hubungan antara error dan sinyal bias terhadap output kontrol relay

**Gambar 3.5.** Ilustrasi hubungan error dan kontrol bias terhadap output kontrol relay

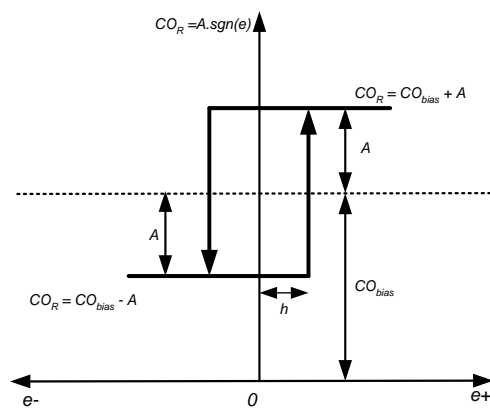
Terkait dengan derau (*noise*) yang pasti muncul pada hasil pengukuran output proses, kontrol relay praktis umumnya dirancang untuk memiliki celah diferensial atau histerisis yang secara fungsional berguna agar output kontrol tidak sensitif terhadap derau tersebut (lihat gambar 3.6).

Lebar histerisis pada kontrol relay itu sendiri sangat tergantung dari level derau yang teridentifikasi pada output proses: semakin besar level derau, maka histerisis pun harus semakin lebar.





(a). Hubungan antara error terhadap output relay

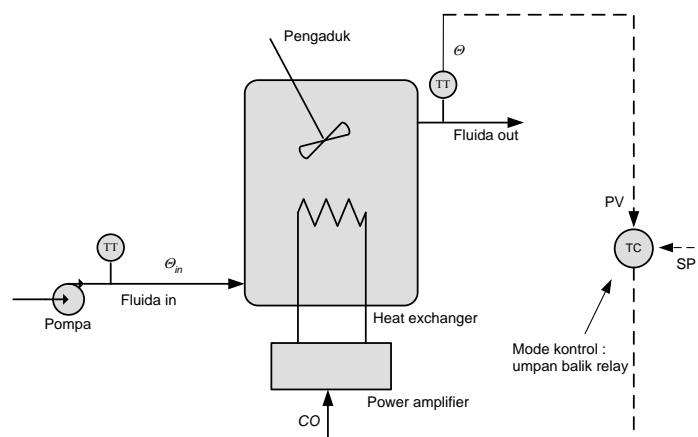


(b). Hubungan antara error dan sinyal bias terhadap output kontrol relay

**Gambar 3.6.** Ilustrasi hubungan error dan kontrol bias terhadap output kontrol relay plus histerisis

- Karakteristik Tanggapan model proses FOPDT Terhadap Kontrol Umpan Balik Relay**

Berdasarkan karakteristik prosesnya, penggunaan metode kontrol relay untuk proses FOPDT pada titik kerjanya secara praktis akan selalu menghasilkan osilasi output yang seimbang disekitar Setpoint yang diberikan.



**Gambar 3.7.** Sistem kontrol Heat Exchanger dengan metode umpan balik relay

Sebagai studi kasus, kita tinjau implementasi kontrol umpan balik relay pada sistem *Heat Exchanger* seperti nampak pada gambar 3.7. Walaupun proses ini pada kenyataannya banyak melibatkan ketidaklinearan, tetapi untuk memudahkan analisis, disini dianggap model proses bersifat linear untuk seluruh daerah kerjanya (jangkauan sinyal kontrol antara 0%-100%), sehingga dengan demikian variabel input-output yang muncul pada model matematis sistem *Heat Exchanger* dapat langsung direpresentasikan dalam bentuk variabel nominalnya (bukan variabel deviasi).

Berdasarkan hasil penyelesaian **contoh soal 1.8 dalam Bab 1**, dinamika temperature fluida output pada sistem *Heat Exchanger* gambar 3.7 tersebut adalah:

$$\Theta(s) = \frac{Ke^{-Ls}}{Ts + 1} CO(s) + \frac{e^{-Ls}}{Ts + 1} \Theta_{in}(s) \quad (3.3)$$

dalam hal ini

$$\text{Konstanta waktu proses: } T = \frac{V\rho}{w}$$

$$\text{Gain statis proses: } K = \frac{K_{ps}}{cw}$$

$$\text{Delay transportasi: } L = \frac{K_L}{w}$$

Jika parameter proses diatas berturut-turut bernilai:

$$V \text{ (Volume tangki)} = 0.2 \text{ m}^3$$

$$\rho \text{ (densitas fluida)} = 1000 \text{ kg/m}^3$$

$$c \text{ (Kalor jenis fluida)} = 4000 \text{ J/(Kg.}^\circ\text{C)}$$

$$K_{ps} \text{ (konstanta kesebandingan antara perubahan panas dengan perubahan sinyal kontrol)} = 800 \text{ W/\% (800.60 J/menit/\% = 48000 J/menit/\%)}$$

$$K_L = 50 \text{ (Kg)}$$

$$w \text{ (laju aliran fluida)} = 20 \text{ kg/menit}$$

Maka model *Heat Exchanger* diatas akan memiliki dinamika seperti nampak pada persamaan 3.4 dibawah (lihat juga diagram bloknya pada gambar 3.8):

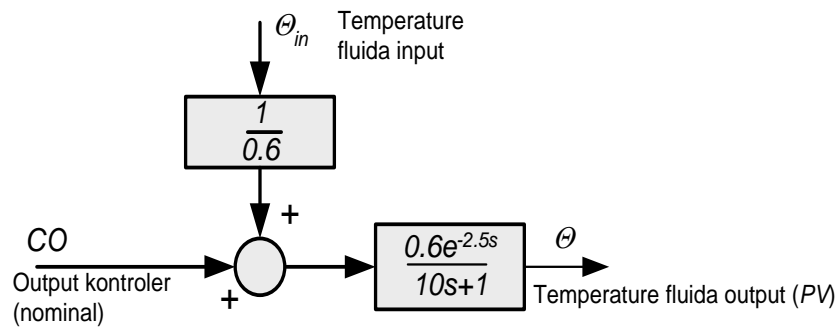
$$\Theta(s) = \frac{0.6e^{-2.5s}}{10s + 1} CO(s) + \frac{e^{-2.5s}}{10s + 1} \Theta_i(s) \quad (3.4)$$

dengan

$\Theta$  = Temperature fluida output nominal (yang terukur sensor)

$\Theta_{in}$  = Temperature fluida input nominal (yang terukur sensor)

$CO$  = Sinyal kontrol nominal



**Gambar 3.8.** Diagram blok Sistem kontrol Heat Exchanger

Jika temperature fluida output tersebut akan dikontrol dengan menggunakan prinsip umpan balik relay, maka langkah pertama yang harus dilakukan adalah mencari terlebih dulu sinyal kontrol bias ( $CO_{bias}$ ) sedemikian sehingga nilai temperature fluida output berada disekitar nilai terharapnya ( $SP$ ).

Besar sinyal kontrol bias yang harus diseting untuk sebuah Setpoint tertentu pada dasarnya dapat dicari dengan memecahkan persamaan 3.4 pada keadaan *steady*-nya:

$$\Theta = 0.6.CO + \Theta_{in} \quad (\text{lihat kembali Bab 1})$$

Misal jika temperature fluida output ( $\Theta$ ) akan diregulasi pada nilai terharap  $40^\circ\text{C}$ , dimana temperature fluida input ( $\Theta_{in}$ ) bernilai  $25^\circ\text{C}$ , maka besar sinyal kontrol ( $CO$ ) yang diperlukan:

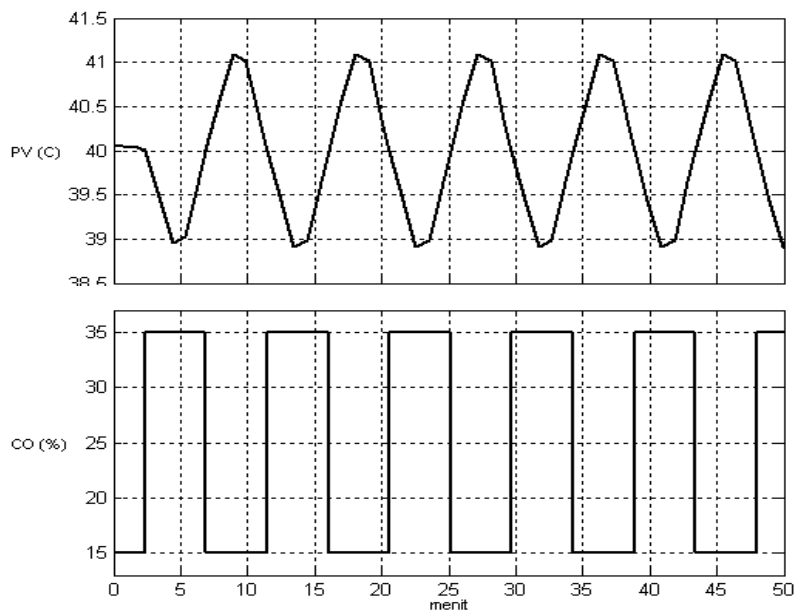
$$40 = 0.6x.CO + 1x25$$

$$CO = \frac{40 - 25}{0.6} = 25\%$$

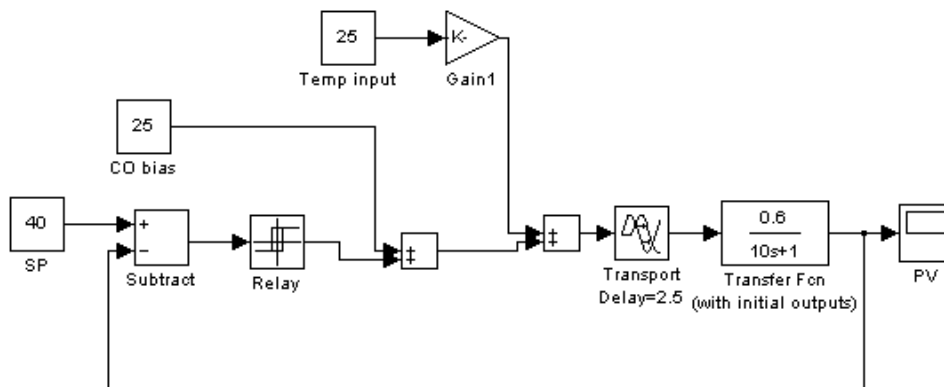
Nilai yang didapat tersebut selanjutnya dijadikan sinyal bias ( $CO_{bias}$ ) pada sistem kontrol umpan balik relay.

Tetapi jika dalam praktek, model matematis proses tidak diketahui secara pasti, maka operator terlebih dulu harus membawa output proses pada titik kerja secara manual (*open loop*). Setelah output proses menetap pada nilai terharapnya, langkah selanjutnya adalah mengubah mode kontroler menjadi mode Auto (*closed loop*). Dengan asumsi kontroler yang digunakan memiliki kemampuan *bumpless transfer*, maka secara otomatis  $CO_{manual}$  tersebut oleh kontroler akan disalin menjadi  $CO_{bias}$ .

Pada sisi lain, agar output proses benar-benar beresilasi di daerah kerjanya secara seimbang, maka pada peralihan mode dari Manual ke Auto (mode kontrol relay), kontroler juga harus memiliki kemampuan menyalin nilai  $PV$  menjadi  $SP$ . Gambar 3.9 memperlihatkan hubungan output kontroler dengan temperature fluida output hasil simulasi sistem *Heat Exchanger* untuk nilai  $SP = 40^\circ\text{C}$  dengan magnitude relay  $A$  sebesar 10% dan temperature fluida input bernilai  $25^\circ\text{C}$ . (lihat juga blok simulink-nya pada gambar 3.10)



**Gambar 3.9.** Hasil simulasi Sistem kontrol Heat Exchanger dengan metode umpan balik relay



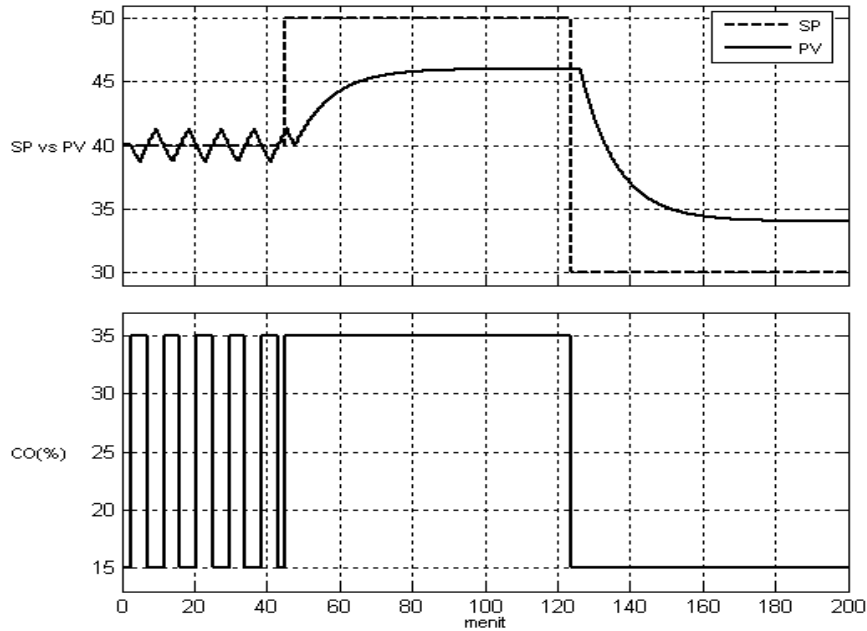
**Gambar 3.10.** Diagram blok Simulink Sistem kontrol Heat Exchanger dengan metode umpan balik relay

Seperti nampak dari hasil simulasi, pengontrolan sebuah proses FOPDT (dalam hal ini sistem *Heat Exchanger*) dengan menggunakan kontrol relay secara praktis akan menyebabkan output proses selalu berosilasi disekitar titik kerja atau Setpoint yang diberikan.

Besar nilai magnitude dan perioda osilasi output proses yang terjadi pada dasarnya sangat tergantung dari besar magnitude output relay yang digunakan: Semakin besar nilai absolute  $A$  maka magnitude dan perioda osilasinya juga semakin besar dan lebar. Dapat dibuktikan bahwa perbandingan antara besar magnitude terhadap lebar perioda osilasi output proses akan memiliki nilai yang konstan.

Sekarang bagaimana tanggapan output proses jika satu saat nilai Setpoint diubah pada nilai yang baru?. Seperti pada kasus kontrol Proporsional, dengan menggunakan sinyal bias yang telah terseting sebelumnya, maka output proses akan/dapat menunjukkan offset

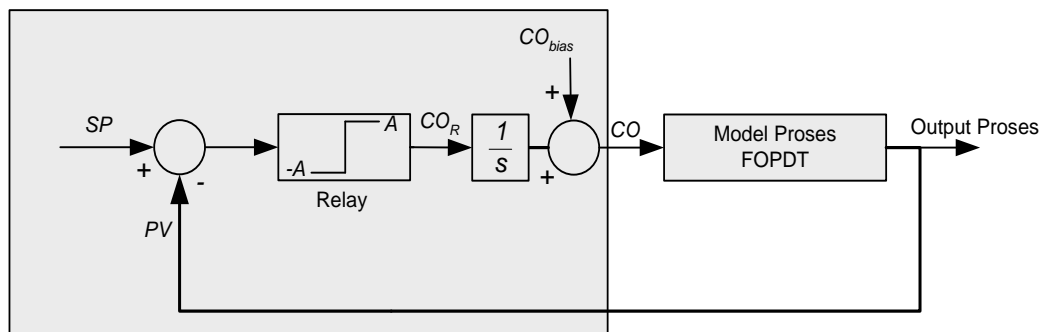
positive atau negative tergantung besar nilai perubahan Setpoint yang diberikan tersebut (lihat ilustrasinya pada gambar 3.11).



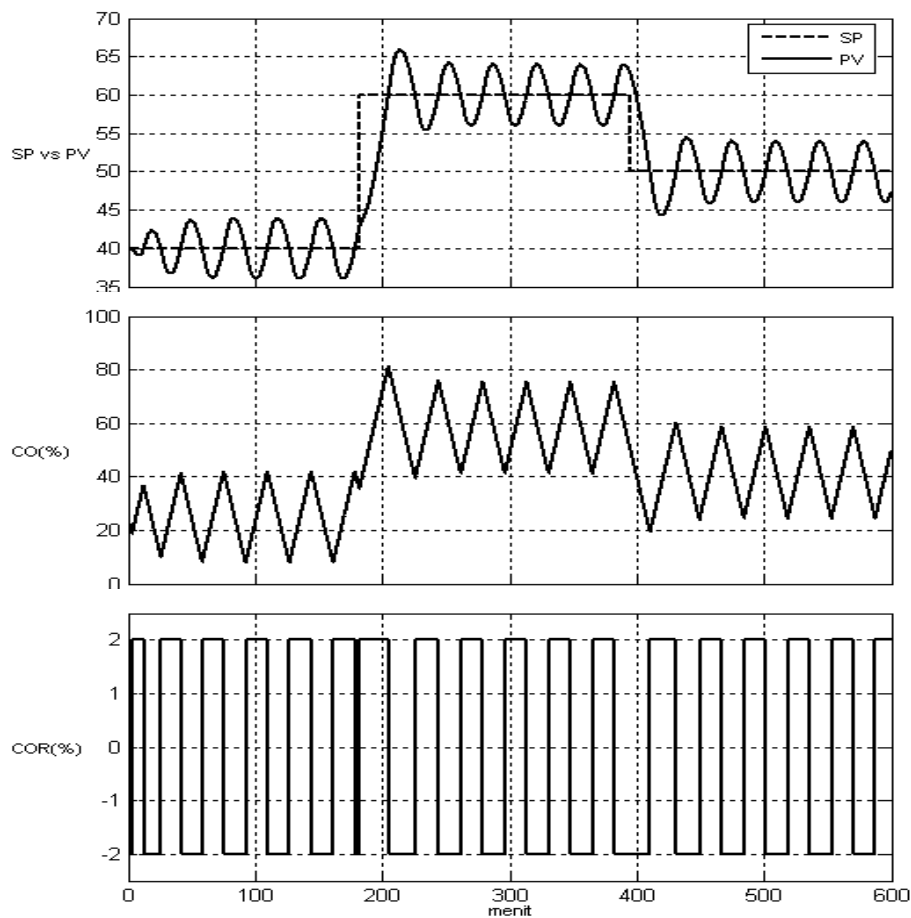
**Gambar 3.11.** Offset yang terjadi pada kasus pengontrolan metode relay untuk model proses FOPDT

Untuk mengatasi offset yang mungkin terjadi, maka setiap kali terjadi perubahan nilai Setpoint, nilai kontrol bias pun perlu diseting ulang lagi.

Secara praktis, Seting ulang sinyal bias dapat dihindari jika pada kontroler yang digunakan terdapat integrator yang disisipkan secara serial dengan relay (lihat gambar 3.12).



**Gambar 3.12** Diagram blok kontrol proses dengan menggunakan metode kontrol relay plus integrator



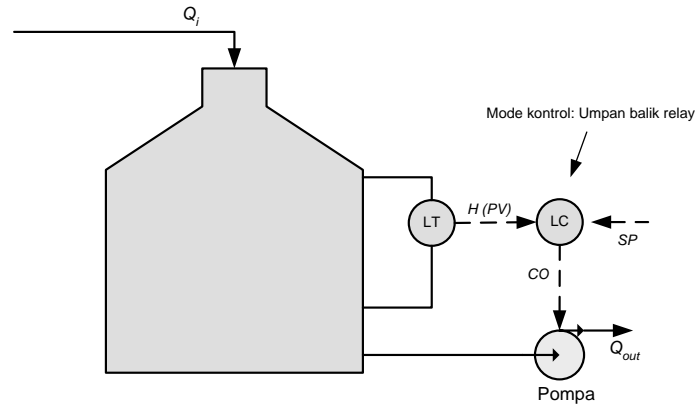
**Gambar 3.13.** Hasil simulasi Sistem kontrol Heat Exchanger dengan metode umpan balik relay plus relay

Satu hal penting yang perlu diperhatikan, dengan adanya integrator pada output relay, maka magnitude serta periode osilasi pada output proses yang dikontrol akan menjadi lebih besar (dan lebih lebar) dibandingkan dengan tanpa integrator. Sehingga dengan demikian, agar magnitude serta periode output proses nilainya masih berada dalam batas toleransi, secara praktis nilai magnitude relay pada kontroler perlu diperkecil. Gambar 3.13 memperlihatkan hasil simulasi hubungan input-output kontrol relay (plus integrator) dengan besar magnitude relay  $A=2$  pada sistem *Heat Exchanger* sebelumnya.

- **Karakteristik Tanggapan model IPDT Terhadap Kontrol Umpan Balik Relay**

Seperti halnya tanggapan output model proses FOPDT, respon output untuk proses IPDT juga akan mengalami osilasi jika proses tersebut dikontrol dengan menggunakan prinsip umpan balik relay.

Sebagai bahan studi kasus, disini akan ditinjau implementasi metode umpan balik relay terhadap salah satu model proses IPDT yang banyak dijumpai di industri yaitu pengontrolan level dengan penggerak pompa penghisap (lihat gambar 3.14).



**Gambar 3.14.** Sistem kontrol level dengan pompa penghisap

Berdasarkan hasil penyelesaian **contoh soal 1.9 pada Bab 1**, persamaan dinamika antara output kontroler dan debit fluida input terhadap level fluida dalam tangki (dengan asumsi karakteristik pompa penghisap tersebut linear) adalah

$$H(s) = -\frac{K^*}{s} e^{-sL} CO(s) + \frac{K_d^*}{s} e^{-sL_d} Q_i(s) \quad (3.5)$$

dalam hal ini

$$\begin{aligned} \text{Gain integratif proses : } K^* &= \frac{K_{out}}{A} \\ \text{Gain integratif gangguan: } K_d^* &= \frac{1}{A} \end{aligned}$$

**Keterangan:**

$A$  = luas penampang,  $K_{out}$  = Konstanta proporsional sinyal kontrol terhadap debit output :  $Q_{out} = K_{out} CO$ .

Dengan menganggap luas penampang tangki  $A = 2 \text{ m}^2$  dan konstanta proporsionalitas,  $K_{out} = 0.24 \text{ m}^3/(\text{menit.}\%)$  Maka persamaan 3.5 dapat ditulis menjadi

$$H(s) = -\frac{0.12}{s} e^{-s} CO(s) + \frac{0.5}{s} Q_i(s) \quad (3.6)$$

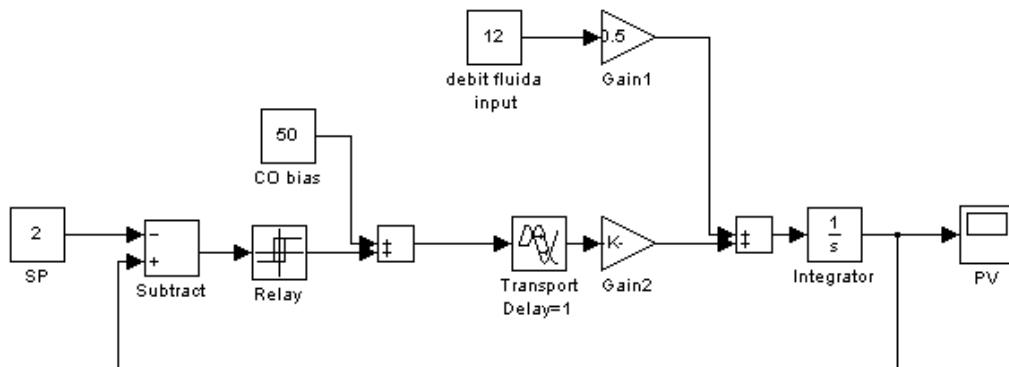
Jika model proses diatas akan dikontrol dengan metode umpan balik relay, maka seperti halnya pada kasus pengontrolan proses FOPDT, langkah pertama yang harus dilakukan adalah membawa output proses pada titik kesetimbangan atau daerah kerja yang diharapkan, baik dilakukan secara manual atau berdasarkan hasil perhitungan.

Untuk besar debit fluida input ( $Q_{in}$ ) tertentu, besar sinyal kontrol penggerak pompa ( $CO$ ) yang diperlukan agar titik kesetimbangan dapat tercapai secara teoritis dapat dicari dari hubungan proporsionalitas  $CO$  vs  $Q_{out}$ :

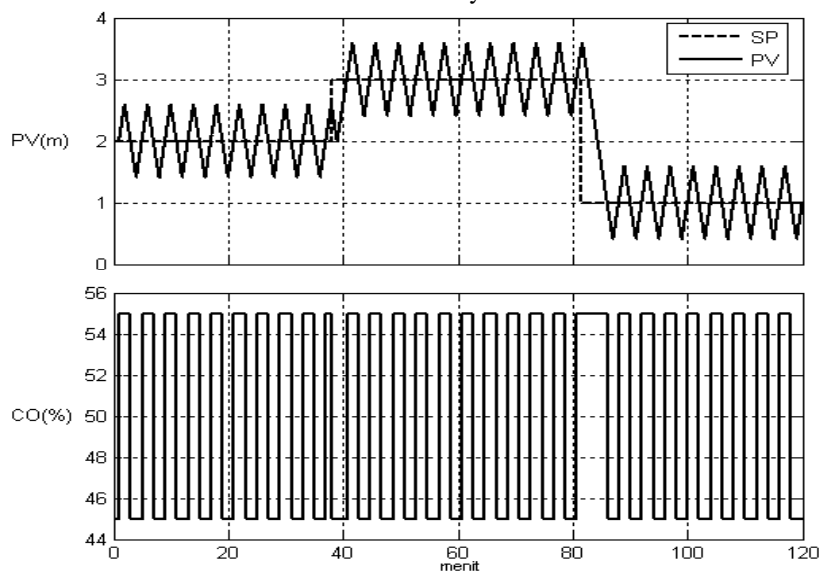
$$CO = \frac{Q_{out}}{K_{out}}$$

Jika misalnya debit fluida input  $Q_{in} = 12 \text{ m}^3/\text{menit}$ , maka titik kesetimbangan dapat tercapai jika debit fluida output juga bernilai sama, yaitu  $Q_{out} = 12 \text{ m}^3/\text{menit}$ , dengan menggunakan relasi proporsionalitas, besar sinyal kontrol yang diperlukan adalah:

$$CO = \frac{Q_{out}}{K_{out}} = \frac{12}{0.24} = 50(\%)$$



**Gambar 3.15.** Diagram blok Simulink Sistem kontrol level fluida dengan metode umpan balik relay



**Gambar 3.16.** Hasil simulasi Sistem kontrol level fluida dengan metode umpan balik relay

Sinyal kontrol yang didapat selanjutnya dijadikan sebagai sinyal bias pada sistem kontrol relay sedemikian sehingga ketika kontroler diubah pada mode Auto, output proses (dalam hal ini level fluida) akan selalu beresilasi disekitar daerah kerja atau titik kesetimbangannya tersebut. Karena secara matematis model proses IPDT ini telah memiliki sifat integrator, maka untuk setiap perubahan nilai  $SP$ , seting ulang sinyal bias secara praktis tidak diperlukan lagi (Sinyal bias hanya perlu diubah jika pada model proses tersebut terjadi perubahan debit fluida input --gangguan).

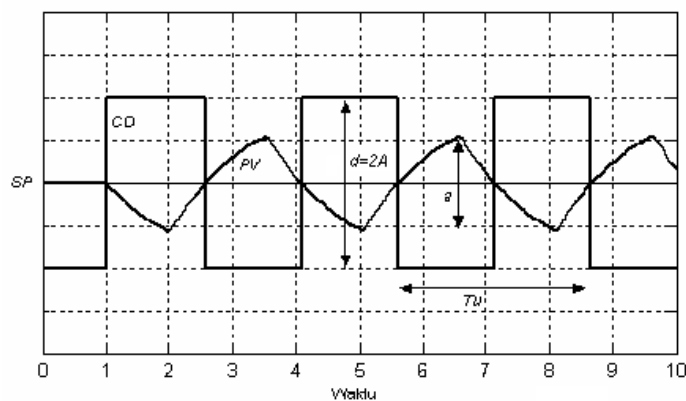
Gambar 3.15 dan 3.16 berturut-turut memperlihatkan diagram Simulink serta salah satu hasil simulasi pengontrolan level fluida dengan sistem kontrol umpan balik relay (dalam simulasi ini magnitudo relay yang digunakan:  $A = 5$  dengan level fluida awal dalam tangki saat kesetimbangan tercapai :  $PV = 2$  meter)



### 3.3.2. Perhitungan parameter kontrol PID berdasarkan informasi magnitude dan perioda osilasi output proses

Seperti telah sedikit disinggung pada subbab sebelumnya, Autotuning dengan metode umpan balik relay pada dasarnya bertumpu pada informasi magnitude dan perioda osilasi yang dihasilkan jika proses (baik model FOPDT maupun model IPDT) dikontrol dengan menggunakan metode umpan balik relay.

Tidak tergantung model proses yang dikontrol, untuk besar magnitude relay yang digunakan, perbandingan antara magnitude relay dan magnitude osilasi output akan menghasilkan nilai yang konstan. Demikian juga perbandingan antara magnitude osilasi output terhadap perioda osilasi: Semakin besar magnitude osilasi, perioda osilasinya juga semakin lebar (lihat ilustrasinya pada gambar 3.17 )



**Gambar 3.17.** Magnitude dan perioda osilasi output proses

Terkait dengan sistem kontrol umpan balik PID, informasi nilai magnitude relay dan nilai magnitude osilasi output proses pada dasarnya dapat dijadikan basis perhitungan besar parameter penguat proporsional maksimum sedemikian sehingga sistem kontrol akan menunjukkan ketidakstabilan. Besar penguat proporsional maksimum ini dikenal juga dengan nama *gain ultimate* yang dirumuskan pada persamaan 3.7 berikut:

$$K_u = \frac{4d}{\pi a} \quad (3.7)$$

dengan  $d = 2A$  (besar sinyal output relay *peak to peak* –lihat gambar 3.17)

$a$  = besar sinyal output proses *peak to peak* –lihat gambar 3.17

Berdasarkan nilai *gain ultimate* (persamaan 3.7) dan perioda osilasi  $T_u$  (lihat gambar 3.17), besar parameter PID secara praktis dapat dihitung dengan menggunakan perumusan atau metode yang telah tersedia. Tabel 3.1 sampai tabel 3.5 berikut memperlihatkan sejumlah metode tuning empiris untuk berbagai struktur PID yang dapat dijumpai dalam sejumlah literature.

**Tabel 3.1.** Metode tuning kontrol Proporsional Integral untuk model proses FOPDT dan IPDT

Metode	$K_P$	$T_I$	Keterangan
Ziegler Nichols	$0.45K_u$	$0.83T_u$	Quarter decay ratio
Hang	$0.25K_u$	$0.2546T_u$	Dominant time delay

Pessen	$0.25K_u$	$0.042K_u \cdot T_u$	Dominant time delay
Shinsky	$0.55K_u$	$0.78T_u$	$L/T=0.2$
Shinsky	$0.48K_u$	$0.47T_u$	$L/T=1$

**Tabel 3.2.** Metode tuning kontrol PID ideal untuk model proses FOPDT dan IPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Pessen-minimum IAE	$0.7K_u$	$0.4T_u$	$0.149T_u$	$0.1 < L/T < 1$
Zhuang and Atherton	$0.509K_u$	$0.051(3.302K_u + 1)T_u$	$0.125T_u$	$0.1 < L/T < 2$
Ziegler-Nichols	$0.6K_u$	$0.5T_u$	$0.12T_u$	-
Corripio	$0.75K_u$	$0.63T_u$	$0.1T_u$	Quarter Decay Ratio
Atkinson Davey	$0.25K_u$	$0.75T_u$	$0.25T_u$	20% overshoot
Mac Millan	$0.5K_u$	$0.5T_u$	$0.125T_u$	-

**Tabel 3.3.** Metode tuning kontrol PIDF untuk model proses FOPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Shinsky	$\frac{KL}{3L - 0.32T_u}$	$T_u \left[ 0.15 \frac{T_u}{L} + 0.05 \right]$	$0.14T_u$	-

**Tabel 3.4.** Metode tuning kontrol PID Serial untuk model proses FOPDT dan IPDT

Metode	$K_P$	$T_I$	$T_D$	Keterangan
Pessen	$0.35K_u$	$\frac{0.25}{T_u}$	$\frac{0.25}{T_u}$	$0.1 < L/T < 1$
	$0.25K_u$	$0.33T_u$	$\frac{0.5}{T_u}$	Optimum servo respon

**Tabel 3.5.** Metode tuning kontrol PI-D untuk model proses FOPDT dan IPDT

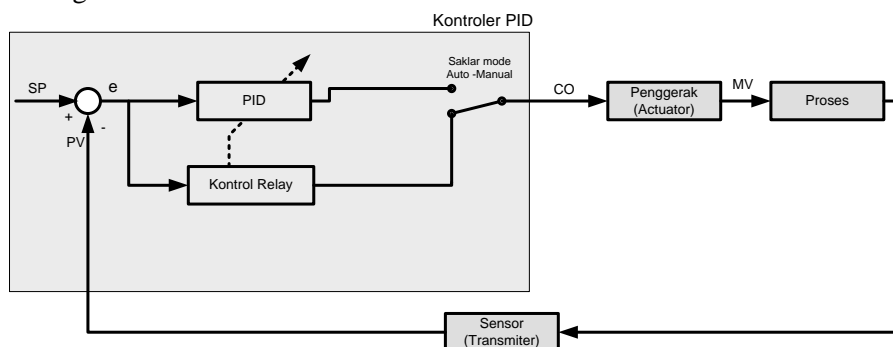
Metode	$K_P$	$T_I$	$T_D$	Keterangan
Shinsky	$0.7692K_u$	$0.48T_u$	$0.11T_u$	$L/T=0.2$
	$0.6993K_u$	$0.42T_u$	$0.12T_u$	$L/T=0.5$
	$0.6223K_u$	$0.38T_u$	$0.12T_u$	$L/T=1$
	$0.6024K_u$	$0.34T_u$	$0.12T_u$	$L/T=2$
Van Doren	$0.75K_u$	$0.625T_u$	$0.1T_u$	-

### 3.3.3. Mekanisme Autotuning Metode umpan balik relay

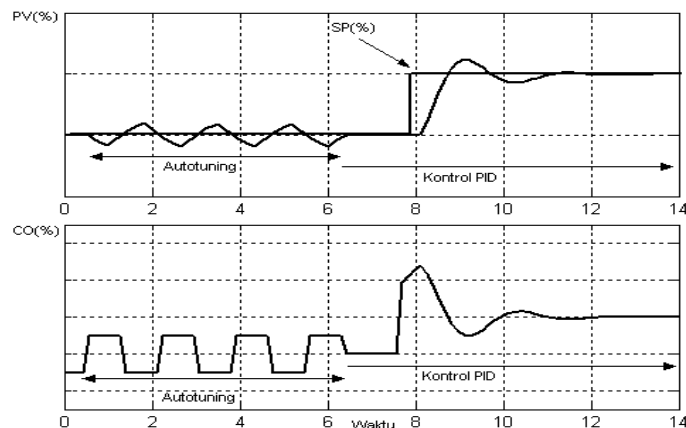
Secara teknis, Autotuning metode umpan balik relay ini hanyalah otomatisasi yang dilakukan modul kontroler atau perangkat lunak tuning terhadap dua hal berikut:

1. Identifikasi gain ultimate ( $K_u$ ) dan perioda osilasi proses ( $T_u$ ) sebagai akibat pengontrolan umpan balik relay
2. Perhitungan parameter kontrol PID dengan menggunakan metode tuning tertentu berdasarkan nilai gain ultimate dan perioda osilasi yang diperoleh pada langkah pertama.

Gambar 3.18 dan 3.19 berikut berturut-turut memperlihatkan ilustrasi mekanisme Autotuning metode relay, serta hubungan  $CO$  dan  $SP$  terhadap  $PV$  sesaat setelah Autotuning diaktifkan.



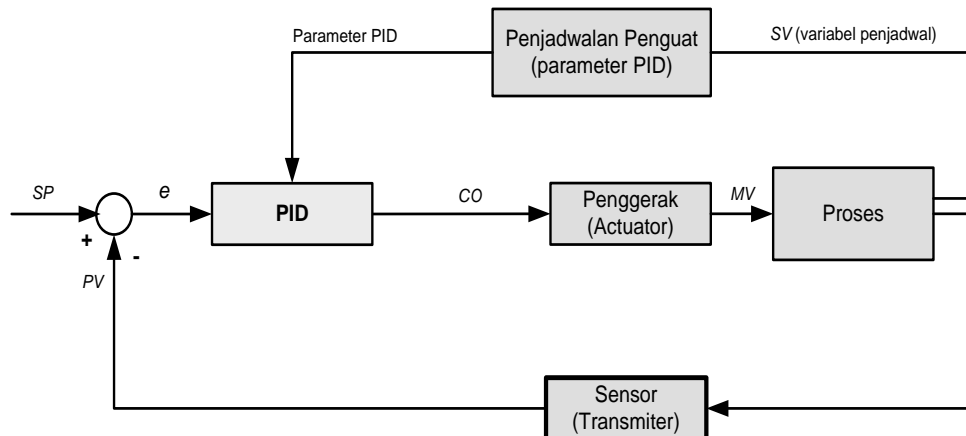
**Gambar 3.18** Ilustrasi metode autotuning umpan balik relay



**Gambar 3.19** Hubungan sinyal kontrol ( $CO$ ) vs variabel proses ( $PV$ ) setelah fasilitas autotuning dijalankan.

### 3.4. Gain Scheduling

Sesuai namanya, Gain scheduling adalah sebuah metode kontrol dimana besar parameter PID yang digunakan oleh modul kontrol dijadwal berdasarkan titik kerja atau kondisi operasi yang dihadapi. Agar setiap saat kontroler mengetahui kondisi proses, maka dalam skema kontrol ini diperlukan pengukuran variabel tambahan (*auxiliaries variable*) yang secara teknis digunakan sebagai variabel penjadwal --*Scheduled variable* (lihat gambar 3.20).



**Gambar 3.20** Diagram Blok Sistem Kontrol Gain Scheduling

Variabel *auxiliary* yang akan dijadikan variabel penjadwal pada dasarnya harus berkorelasi atau berhubungan dengan titik kerja atau kondisi proses yang dikontrol. Variabel tersebut dapat saja berupa output prosesnya itu sendiri atau variabel input lain yang secara langsung mempengaruhi dinamika proses.

Dalam bentuknya yang sederhana, metode kontrol Gain Scheduling ini dapat direalisasikan dalam sebuah table tengok (*look up table*) yang berisi himpunan parameter PID untuk berbagai kondisi yang mungkin terjadi selama pengontrolan proses berlangsung. Tabel 3.6 berikut memperlihatkan ilustrasi tabel penjadwal pada skema kontrol Gain Scheduling.

Tabel 3.6. Himpunan parameter PID untuk berbagai kondisi proses

	$K_P$	$T_I$	$T_D$
<b>Kondisi 1</b>	$K_{P1}$	$T_{I1}$	$T_{D1}$
<b>Kondisi 2</b>	$K_{P2}$	$T_{I2}$	$T_{D2}$
<b>Kondisi --</b>	--	--	--
<b>Kondisi n</b>	$K_{Pn}$	$T_{In}$	$T_{Dn}$

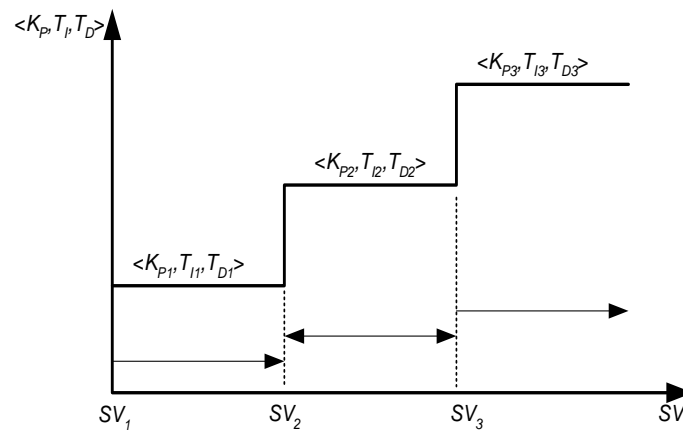
Kondisi-kondisi yang mencerminkan dinamika proses secara praktis dapat diketahui secara langsung dari nilai variabel *auxiliary* atau variabel penjadwal yang terukur sensor. Dalam perancangan awal, kita terlebih dulu harus menentukan jangkauan nilai variabel penjadwal (SV) untuk setiap kondisi dinamika proses yang sedang dihadapi: Proses berada dalam satu kondisi tertentu, misal jika nilai variabel penjadwal yang terukur ada dalam range yang telah ditentukan sebelumnya. Tabel 3.7 berikut memperlihatkan salah satu contoh rancangan tabel penguat untuk tiga kondisi proses berdasarkan jangkauan nilai variabel penjadwal:

Tabel 3.7. Tiga himpunan parameter PID untuk tiga kondisi proses

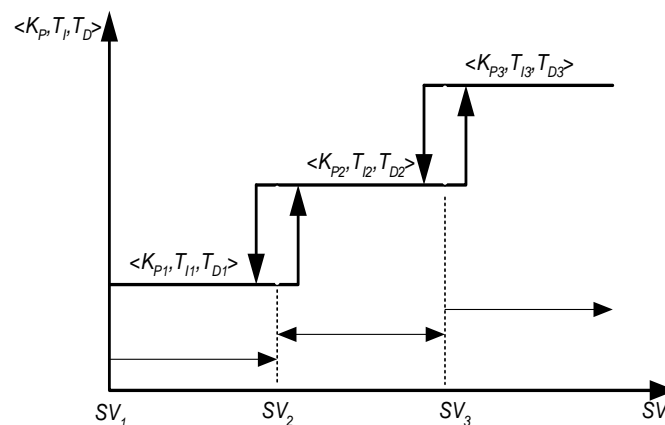
	$K_P$	$T_I$	$T_D$
<b>Kondisi 1 :</b>	$K_{P1}$	$T_{I1}$	$T_{D1}$

$SV_1 < SV < SV_2$			
<b>Kondisi 2 :</b> $SV_2 < SV < SV_3$	$K_{P2}$	$T_{I2}$	$T_{D2}$
<b>Kondisi 3:</b> $SV > SV_3$	$K_{P3}$	$T_{I3}$	$T_{D3}$

Berdasarkan tabel rancangan penguat tersebut, nampak bahwa jika satu saat variabel penjadwal ( $SV$ ) yang terukur nilainya diantara  $SV_1$  dan  $SV_2$ , maka parameter PID yang digunakan oleh kontroler berturut-turut adalah  $K_{P1}$ ,  $T_{I1}$ ,  $T_{D1}$ , sedangkan jika nilai variabel penjadwal berada diantara  $SV_2 - SV_3$ , maka parameter PID yang digunakan berturut-turut  $K_{P2}$ ,  $T_{I2}$ ,  $T_{D2}$ , dan seterusnya. Dalam bentuk grafik, tabel 3.7 diatas dapat diilustrasikan oleh gambar 3.21 berikut:



**Gambar 3.21** Ilustrasi hubungan nilai variable penjadwal dengan parameter PID



**Gambar 3.22** penggunaan histeris pada variable penjadwal

Karena dalam pengukuran variabel penjadwal secara praktis akan selalu menghasilkan derau, maka untuk menghindari peralihan parameter kontrol yang disebabkan oleh derau tersebut, dalam perancangan awal perlu ditambahkan histeris pada setiap peralihan parameter (lihat gambar 3.22).

Terkait dengan implementasi skema kontrol gain scheduling pada proses pengontrolan, dalam subbab ini akan ditinjau penerapan skema kontrol gain scheduling untuk dua problem berikut:

1. Pengontrolan sistem *Heat exchanger* dengan beberapa kondisi beban yang berbeda
2. Pengontrolan level fluida pada tangki dengan penampang nonlinear

### 3.4.1 Studi Kasus Gain Scheduling pada Sistem Heat Exchanger

Dalam sebuah sistem *Heat Exchanger*, laju produksi pada dasarnya terkait langsung dengan debit fluida input yang akan diproses: semakin tinggi laju produksi maka artinya debit fluida input semakin besar.

Secara matematis, perbedaan debit fluida input tersebut akan berpengaruh langsung terhadap nilai parameter model proses yang akan dikontrol. Dengan menggunakan model dan nilai parameter yang sama seperti pada sistem *Heat Exchanger* gambar 3.7 (kecuali nilai debit input yang dapat berubah), dinamika temperature fluida output pada persamaan 3.3. sekarang merupakan fungsi dari debit aliran fluida input (lihat persamaan 3.8 dibawah)

$$\Theta(s) = \frac{\frac{12}{w} e^{-\frac{50}{w}s}}{\frac{200}{w}s + 1} CO(s) + \frac{\frac{e^{-\frac{50}{w}s}}{200}}{\frac{w}{s} + 1} \Theta_{in}(s) \quad (3.8)$$

Berdasarkan persamaan 3.8, terlihat jelas bahwa semakin besar debit fluida input maka gain statis proses  $--K$  akan semakin kecil (proses semakin tidak sensitif), Konstanta waktu proses  $--T$  akan semakin kecil (respon proses semakin cepat) dan delay transportasi  $--L$  akan semakin kecil.

Ditinjau dari sudut pengontrolan umpan balik, perubahan debit tersebut pada dasarnya merupakan salah satu bentuk gangguan pada proses pengontrolan.

Sebagai studi kasus, anggap sistem *Heat Exchanger* dalam operasinya memiliki tiga laju produksi yang mungkin terjadi, masing-masing : Laju produksi tinggi (25 kg/men), laju produksi menengah (15 kg/men), dan laju produksi rendah (10 kg/men). Untuk besar laju produksi tersebut, dinamika prosesnya ditunjukkan berturut-turut oleh persamaan 3.9 sampai 3.11 dibawah (persamaan-persamaan tersebut diperoleh dengan mensubstitusikan setiap laju produksi pada persamaan 3.8):

- **Dinamika temperature output untuk laju produksi :25 kg/menit:**

$$\Theta(s) = \frac{0.48e^{-2s}}{8s + 1} CO(s) + \frac{e^{-2s}}{8s + 1} \Theta_{in}(s) ; K=0.48, L=2, T=8 \quad (3.9)$$

- **Dinamika temperature output untuk Laju produksi :15 kg/menit:**

$$\Theta(s) = \frac{0.8e^{-3.3s}}{13.3s + 1} CO(s) + \frac{e^{-3.3s}}{13.3s + 1} \Theta_{in}(s) ; K=0.8, L=3.3, T=13.3 \quad (3.10)$$

- **Dinamika temperature output untuk Laju produksi : 10 kg/menit:**

$$\Theta(s) = \frac{1.2e^{-5s}}{20s+1}CO(s) + \frac{e^{-5s}}{20s+1}\Theta_{in}(s); K=1.2, L=5, T=20 \quad (3.11)$$

Apabila sistem *Heat Exchanger* diatas akan dikontrol dengan menggunakan kontrol umpan balik PID, maka nilai parameter PID yang optimal untuk debit fluida input tertentu, bisa jadi akan menghasilkan unjuk kerja yang tidak memuaskan jika digunakan untuk besar laju fluida yang berbeda: semakin besar perbedaan laju fluida input, unjuk kerja pengontrolan akan semakin buruk.

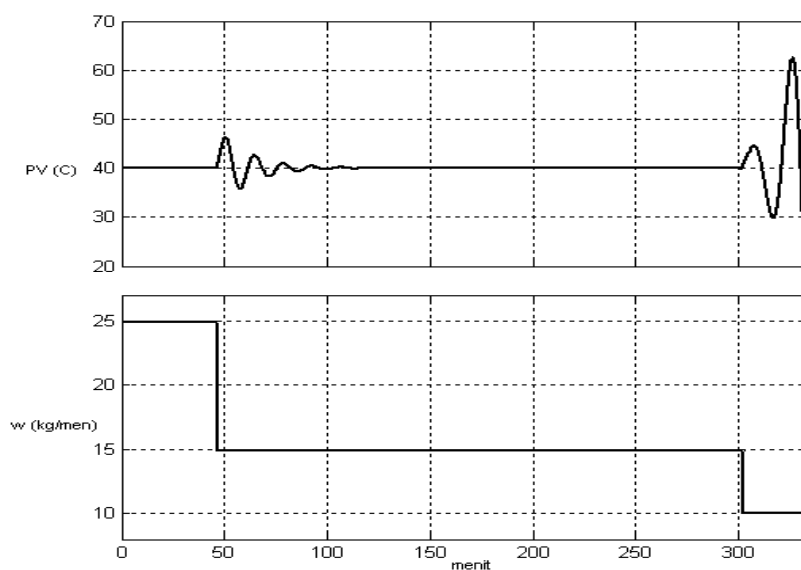
Hal tersebut terkait dengan besar nilai parameter PID optimal yang diperoleh untuk setiap kasus laju produksi akan relatif berbeda: semakin besar perbedaan laju produksi, maka perbedaan parameter PID optimalnya pun akan semakin signifikan. Tabel 3.8 dibawah ini memperlihatkan parameter PID yang diperoleh dengan menggunakan metode tuning Astrom-Haglung untuk PID ideal berdasarkan informasi parameter proses untuk masing-masing besar laju produksi (lihat kembali **bab 2**)

**Tabel 3.8.** Tiga himpunan parameter PID untuk tiga kondisi proses

Astrom-Haglung	$K_p = \frac{0.94T}{KL}$	$T_I = 2.L$	$T_D = 0.5L$
$w=25$ kg/men	7.83	4	1
$w=15$ kg/men	4.73	6.6	1.65
$w=10$ kg/men	3.13	10	2.5

Sebagai contoh, anggap pada laju produksi awal, fluida input nominal yang mengalir memiliki laju  $w=25$  kg/men. Maka berdasarkan tabel 3.8, untuk laju fluida tersebut parameter PID ideal optimal menurut Astrom-Haglung berturut-turut adalah  $K_p = 7.83$ ,  $T_I = 4$  dan  $T_D = 1$ .

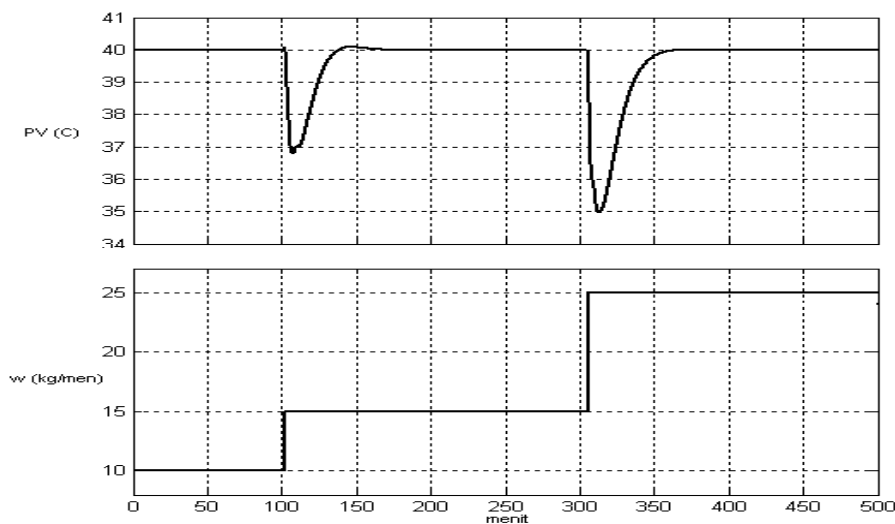
Dengan menggunakan nilai parameter PID diatas, unjuk kerja pengontrolan untuk setiap perubahan besar laju produksi dapat dilihat pada gambar 3.23 hasil simulasi.



**Gambar 3.23** Pengaruh pengecilan laju produksi dari nilai awal 25 kg/men terhadap dinamika output proses

Seperti nampak pada gambar 3.23, perubahan laju produksi dari 25 kg/men ke 15 kg/men akan menyebabkan respon mengalami osilasi yang relative masih teredam, tetapi ketika laju produksi diperkecil pada nilai 10 kg/men (atau mungkin lebih kecil lagi) ternyata output proses menjadi tidak stabil. Mengapa demikian?. Hal ini pada dasarnya terkait dengan nilai paramter PID yang didapat sudah tidak optimal lagi untuk laju produksi yang kecil tersebut (Dalam hal ini besar penguat proporsional untuk laju produksi 10 kg/men nilainya terlalu besar, sedangkan disisi lain parameter time integrator nilainya terlalu kecil). Untuk menghindari kejadian seperti diatas, secara praktis ketika laju produksi diubah maka besar parameter PID-nya pun harus disesuaikan.

Jika dalam pengontrolan yang dipentingkan adalah kestabilan, akan lebih aman jika parameter PID, di-tuning pada debit fluida terkecil (dalam hal ini  $w=10$  kg/men). Gambar 3.24 dibawah memperlihatkan dinamika output proses terhadap perubahan laju produksi dimana PID yang digunakan diseting untuk laju produksi  $w=10$  kg/men ( $K_p=3.13$ ,  $T_i=10$ ,  $T_d=2.5$  –lihat tabel 3.8)



**Gambar 3.24** Dinamika output proses terhadap pembesaran laju produksi dimana parameter PID diseting untuk laju produksi minimal

Seperti nampak pada gambar 3.24, dengan menggunakan nilai PID optimal untuk laju produksi  $w=10$  kg/men (laju produksi minimal), respon proses dijamin stabil untuk setiap perubahan positive pada laju produksi. Tetapi walaupun demikian, dibandingkan dengan gambar 3.23, respon output proses untuk setiap perubahan laju produksi akan menghasilkan dinamika yang semakin lambat.

Agar tidak dilakukan tuning ulang untuk setiap perubahan beban yang relative besar, maka kita dapat menjadwal besar parameter PID untuk setiap besar laju produksi yang berbeda. Jika kontroler yang digunakan memiliki fasilitas Gain Scheduling, penjadwalan tersebut pada dasarnya dapat dilakukan secara otomatis oleh kontroler.

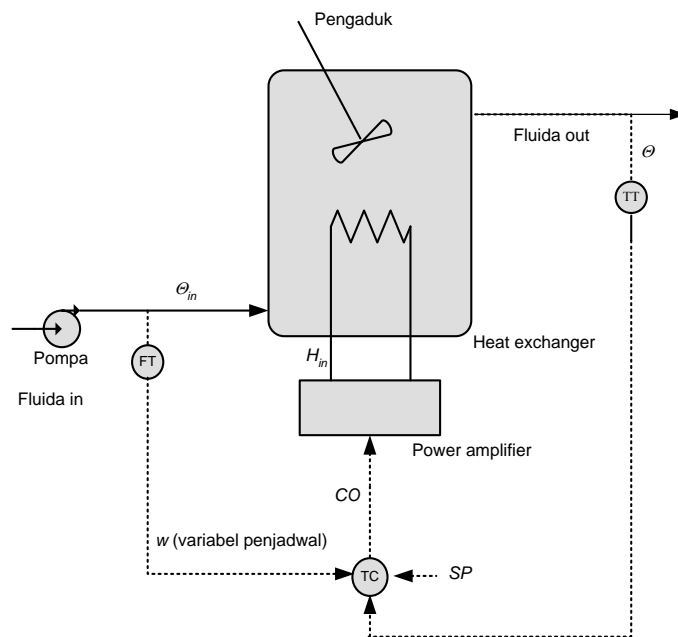
Dalam kasus pengontrolan diatas, nampak jelas bahwa parameter kontrol PID untuk setiap saat, nilainya dapat dijadwal berdasarkan besar laju debit fluida input. Besarnya parameter PID yang digunakan untuk setiap kondisi secara teknis dapat mengacu pada tabel 3.9 berikut:

**Tabel 3.9.** Look Up Table pada Gain Scheduling

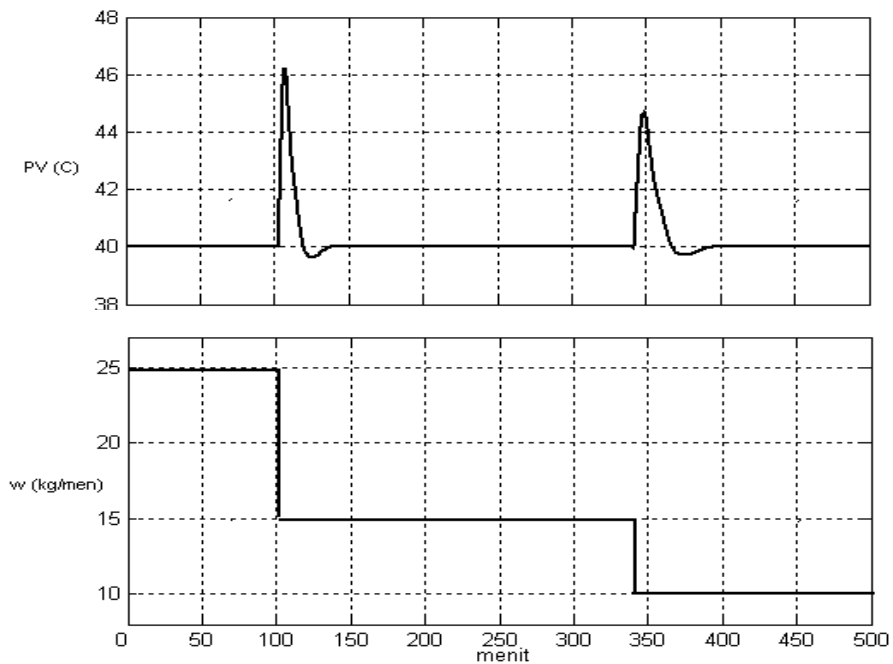
Laju Produksi (Kg/men)	$K_p$	$T_I$	$T_D$
------------------------	-------	-------	-------



$w \geq 25$	7.83	4	1
$15 \leq w < 25$	4.73	6.6	1.65
$10 \leq w < 15$	3.13	10	2.5



**Gambar 3.25** Topologi kontrol Gain Scheduling pada sistem Heat Exchanger

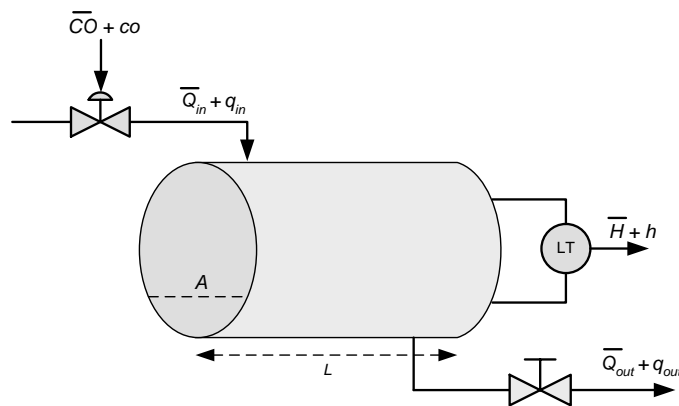


**Gambar 3.26** Dinamika output proses terhadap perubahan laju produksi dengan menggunakan topologi kontrol Gain Scheduling

Gambar 3.25 dan 3.26 berturut-turut memperlihatkan diagram instrumentasi proses dan hasil simulasi sistem kontrol *Heat Exchanger* dengan menggunakan metode Gain Scheduling ( Dalam simulasi ini, parameter PID optimal untuk setiap laju produksi, besarnya merujuk pada tabel 3.9).

### 3.4.2 Studi Kasus Gain Scheduling pada Kontrol Level dengan Penampang Tanki Non Linear

Tinjau sistem kontrol level pada tangki dengan penampang non linear seperti nampak pada gambar 3.27 berikut:



**Gambar 3.27** Sistem kontrol level pada tangki dengan penampang non linear

Dengan menganggap posisi *control valve* berdekatan dengan tangki penampung, maka model matematis perubahan ketinggian ( $h$ ) terhadap perubahan sinyal kontrol ( $co$ ) disekitar nilai *steady* nominalnya dapat dicari dengan menggunakan Hukum kesetimbangan massa berikut:

Laju akumulasi massa pada tangki = laju massa input – laju massa output

Atau dapat ditulis secara matematis :

$$\frac{d}{dt}(\rho Ah(t)) = \rho q_{in} - \rho q_{out} \quad (3.12)$$

dengan:

- $q_{in}$  = perubahan laju aliran fluida input disekitar nilai nominalnya (m<sup>3</sup>/det)
- $q_{out}$  = perubahan laju aliran fluida output disekitar nilai nominalnya(m<sup>3</sup>/det)
- $\rho$  = densitas fluida(kg/m<sup>3</sup>)
- $h$  = Perubahan ketinggian fluida dalam tangki (m)
- $A$  = luas penampang tangki (m<sup>2</sup>) pada ketinggian yang ditinjau

Seperti nampak dalam gambar 3.27, luas penampang tangki untuk setiap titik operasi pada dasarnya akan bersifat variabel (berubah) dan nilainya sangat tergantung pada level fluida *steady* awalnya. Untuk bentuk plant tangki tersebut, luas penampang memiliki relasi seperti nampak pada persamaan 3.13 berikut:

$$A(H) = 2L\sqrt{R^2 - (R - H)^2} \quad (3.13)$$

dengan

$H$ : ketinggian nominal fluida

Pada sistem tangki penampung, perubahan debit aliran fluida output akan berbanding lurus dengan perubahan ketinggian fluida pada tangki ( $h$ ) :

$$q_{out} = K_{out}h(t) \quad (3.14)$$

dengan :

$K_{out}$  = konstanta proporsionalitas yang harganya tergantung bukaan valve output

Jika *control valve* yang digunakan bertipe ATO, maka perubahan debit fluida input akan proporsional terhadap besar perubahan sinyal kontrol ( $co$ ) penggerak *valve*:

$$q_{in} = K_{in}co \quad (3.15)$$

dengan mensubstitusikan persamaan (3.13), (3.14) dan (3.15) pada persamaan (3.12) serta dengan melakukan sedikit langkah penyederhanaan, maka akan diperoleh persamaan diferensial berikut

$$\frac{A(H)}{K_{out}} \frac{dh(t)}{dt} = -h(t) + \frac{K_{in}}{K_{out}} co(t) \quad (3.16)$$

Atau dapat ditulis dalam bentuk fungsi alih

$$\frac{h(s)}{co(s)} = \frac{K}{Ts + 1} \quad (3.17)$$

dalam hal ini

$$K = \text{gain statis proses (nilainya konstan)} ; K = \frac{K_{in}}{K_{out}}$$

$$T = \text{Konstanta waktu (nilainya bervariasi tergantung level)}; T = \frac{A(H)}{K_{out}}$$

Berdasarkan persamaan 3.17 tersebut, terlihat jelas bahwa konstanta waktu ( $T$ ) untuk *plant level* yang dibahas ini akan memiliki nilai yang bersifat variabel yaitu tergantung pada level ketinggian nominalnya. Tabel 3.10 dibawah memperlihatkan nilai gain statis dan konstanta waktu proses untuk dua level ketinggian yang berbeda pada tangki nonlinear dengan nilai parameter berikut:

$$K_{out} = 0.5 \text{ m}^2/\text{s}$$

$$K_{in} = 0.005 \text{ m}^3/\text{s}\%$$

$$R \text{ (jari-jari tangki)} = 0.5 \text{ meter}$$

$$L \text{ (panjang tangki)} = 1 \text{ meter}$$

Tabel 3.10. Nilai parameter proses level fluida untuk dua nilai ketinggian

Ketinggian nominal -H (meter)	$K$	$T$
0.1	0.01	1.2
0.5	0.01	2

0.9	0.01	1.2
-----	------	-----

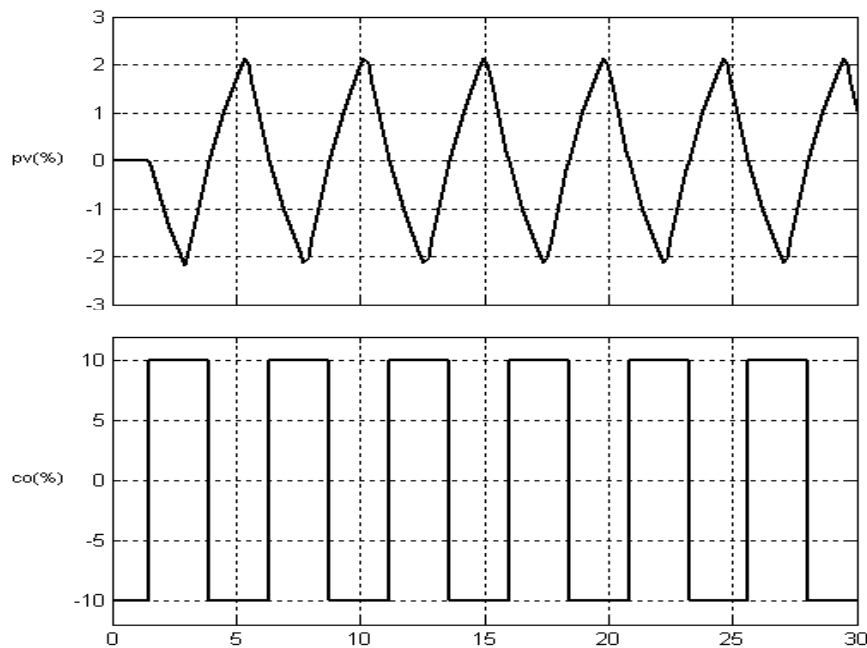
Sehingga dengan demikian agar sistem kontrol level menghasilkan unjuk kerja yang optimal untuk berbagai ketinggian, maka parameter PID yang digunakannya pun harus disesuaikan dengan level ketinggian nominalnya. Dalam hal ini, salah satu teknik yang dapat digunakan adalah dengan menggunakan metode kontrol gain scheduling dimana variabel penjadwal yang digunakan adalah level fluida itu sendiri.

### 3.5. Contoh Soal dan Penyelesaian

#### Contoh 3.1

#### Perhitungan parameter PI optimal berdasarkan informasi osilasi proses

Gambar 3.28 dibawah memperlihatkan hubungan perubahan sinyal kontrol umpan balik relay terhadap output proses tertentu. Berdasarkan informasi periode dan magnitude osilasi output pada gambar, carilah parameter PI optimal seperti yang tersaji pada tabel 3.1 untuk pengontrolan proses tersebut



**Gambar 3.28** Dinamika output proses tertentu dengan kontrol relay

#### Penyelesaian.

Seperti yang telah dibahas sebelumnya, parameter PI (atau PID) untuk pengontrolan proses secara praktis dapat diperoleh berdasarkan dua buah informasi respon frekuensi: periode osilasi output proses -- $T_u$  dan gain ultimate proses -- $K_u$ .

Berdasarkan grafik, nampak bahwa lama periode osilasi untuk proses tersebut :  $T_u = 5$  menit, dengan besar sinyal output relay *peak to peak* -- $d$  dan besar sinyal output proses *peak to peak* -- $a$  berturut-turut sama dengan 10 dan 4, sehingga dengan demikian besar *gain ultimate* --nya:

$$K_u = \frac{4 \times 20}{3.14 \times 4} = 6.4$$

Dengan mensubsitusikan nilai  $T_u$  dan  $K_u$  diatas pada persamaan-persamaan tuning tabel 3.1, parameter PI optimal untuk pengontrolan proses tersebut akan memiliki nilai seperti nampak pada tabel 3.11 dibawah

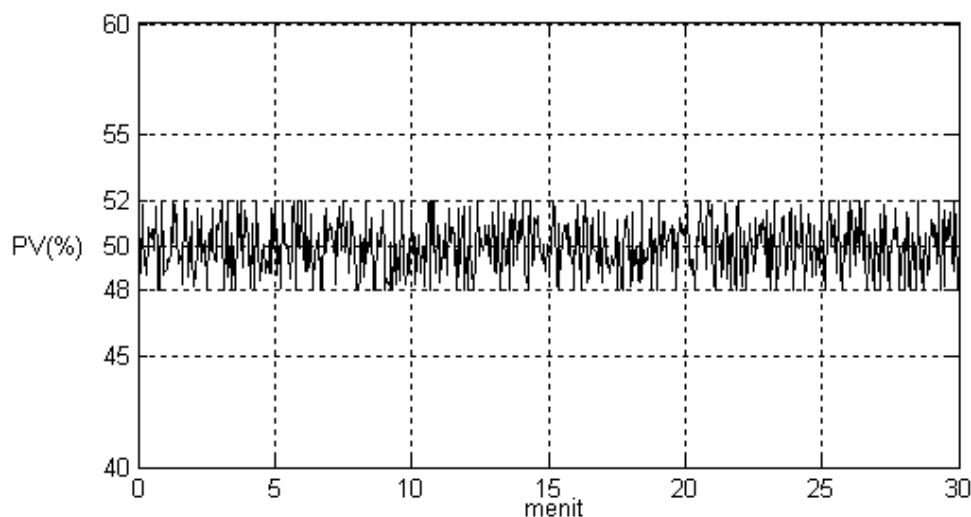
**Tabel 3.11.** Metode tuning kontrol PI untuk model proses pada contoh soal 3.1

Metode	$K_P$	$T_I$	Keterangan
Ziegler Nichols	2.88	4.15	Quarter decay ratio
Hang	1.6	1.273	Dominant time delay
Pessen	1.6	1.344	Dominant time delay
Shinsky	3.52	3.9	$L/T=0.2$
Shinsky	3.072	2.35	$L/T=1$

### Contoh 3.2

#### Perhitungan histerisis

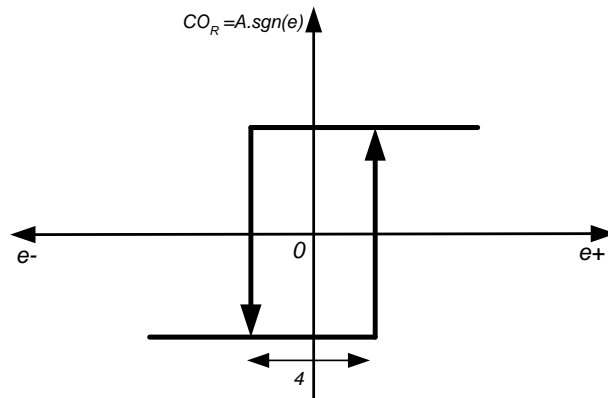
Berapakah histerisis yang harus digunakan pada kontrol umpan balik relay untuk proses yang memiliki karakteristik output berderau seperti nampak dalam gambar 3.29 berikut:



**Gambar 3.29** karakteristik output berderau

#### Penyelesaian

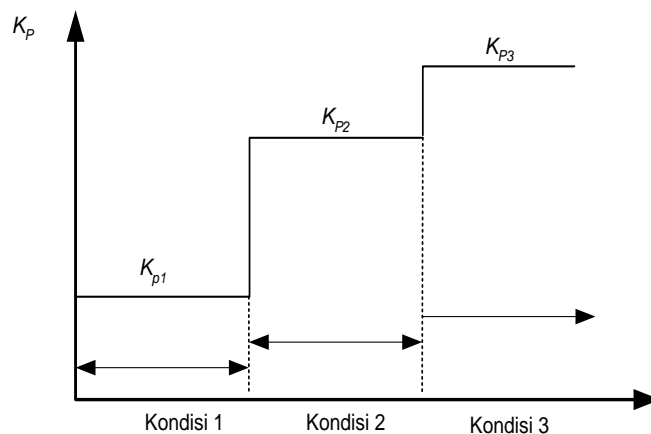
Berdasarkan pengamatan dari grafik pada gambar 3.29, nampak bahwa level derau untuk output proses tersebut bernilai kurang lebih sebesar 4%. sehingga dengan demikian lebar histerisis minimal yang harus digunakan pada kontrol output relay tersebut harus lebih besar dari level derau yang terdeteksi. (lihat ilustrasinya pada gambar 3.30).



**Gambar 3.30** Ilustrasi lebar histerisis yang harus digunakan pada kontrol relay terkait dengan derau pengukuran

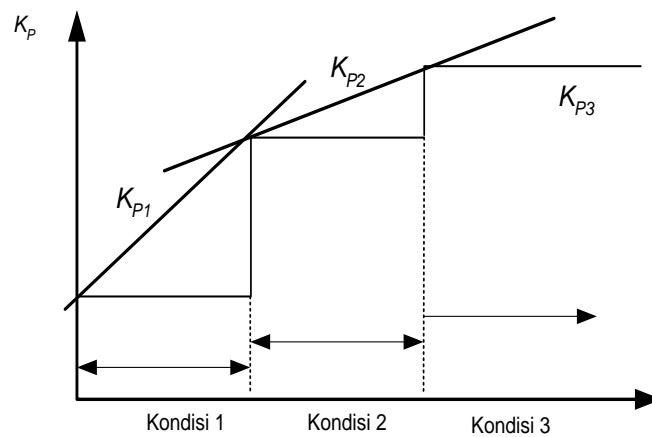
### Contoh Soal 3.3 Persamaan Gain Scheduling

Salah satu kelemahan realisasi Gain Scheduling dengan menggunakan tabel tengok seperti yang telah dibahas sebelumnya adalah terjadinya hentakan sinyal kontrol sebagai akibat perbedaan parameter kontrol yang bersifat mendadak pada setiap perubahan kondisi operasi (lihat ilustrasinya pada perubahan gain proporsional dalam skema kontrol Gain Scheduling gambar 3.31 dibawah).



**Gambar 3.31** Perubahan nilai parameter kontrol (dalam hal ini gain proporsional) secara mendadak akan mengakibatkan hentakan pada sinyal kontrol

Agar secara praktis tidak terjadi hentakan sinyal kontrol tersebut, maka secara teknis metode kontrol Gain Scheduling dapat direalisasikan dengan menggunakan teknik interpolasi seperti diilustrasikan pada gambar 3.32.



**Gambar 3.32** Teknik interpolasi untuk mendapatkan parameter kontrol PID

dengan menggunakan teknik interpolasi diatas, parameter kontrol ( $K_p$ ,  $T_I$  dan  $T_D$ ) dapat dicari secara otomatis berdasarkan fungsi linear yang merelasikan variabel penjadwal dengan parameter kontrol pada setiap kondisi yang dihadapi, atau secara matematis dapat ditulis:

$$< K_p, T_I, T_D > = f(SV)$$

### Pertanyaan

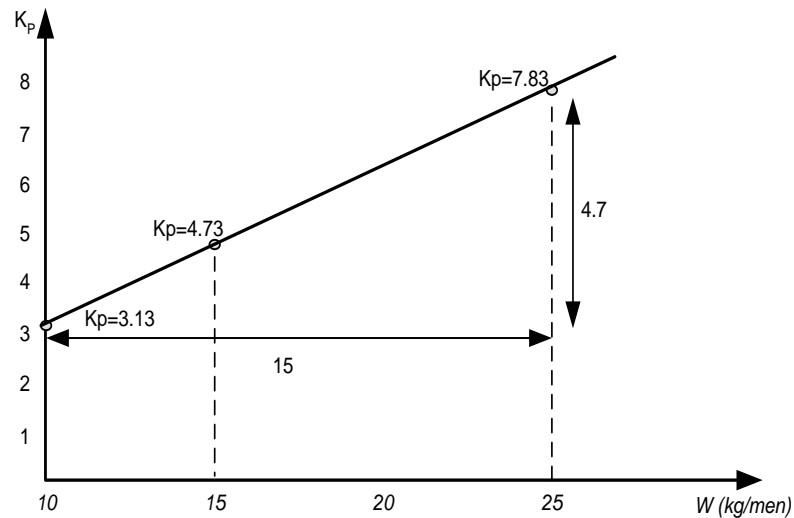
Dengan menggunakan teknik interpolasi, carilah nilai gain proporsional pada studi kasus Gain Scheduling sistem *Heat Exchanger* sebelumnya sebagai fungsi linear dari debit fluida input ( $w$ )

### Penyelesaian

Dengan mengacu tabel 3.9, diketahui Gain Proporsional untuk setiap kondisi laju produksi:

Laju Produksi (Kg/men)	$K_p$
$w \geq 25$	7.83
$15 \leq w < 25$	4.73
$10 \leq w < 15$	3.13

Dalam bentuk grafik, tabel diatas dapat digambarkan:



**Gambar 3.33** Nilai gain penguat ( $K_p$ ) sebagai fungsi linear laju produksi

Dengan mengacu gambar 3.33 diatas, terlihat bahwa besar nilai gain proporsional untuk setiap besar laju produksi pada jangkauan nilai :  $10 \leq w \leq 25$  memiliki hubungan yang relative linear. Sehingga nilai  $K_p$  untuk laju fluida pada jangkauan tersebut dapat dicari berdasarkan fungsi linear berikut:

$$K_p = \frac{\Delta K_p}{\Delta w} (w - w_{\min}) + K_{p \min}$$

dimana dalam hal ini  $K_{p \min}$  adalah nilai gain proporsional untuk laju produksi paling minimal (lihat gambar 3.33).

dengan mensubsitusikan nilai pada grafik, maka persamaan gain proporsional untuk jangkauan  $10 \leq w \leq 25$  adalah:

$$K_p = \frac{4.7}{15} (w - 10) + 3.13$$

### 3.5 Ringkasan

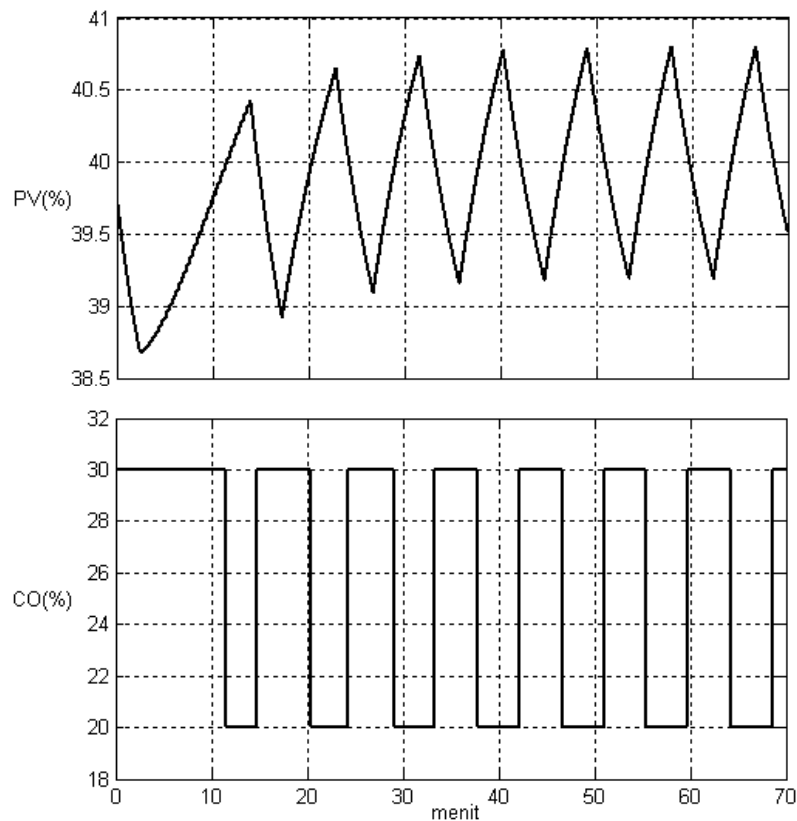
- Autotuning merupakan salah satu fasilitas modul kontrol dimana parameter PID yang tepat dapat diperoleh secara otomatis.
- Berdasarkan sinyal eksitasinya, ada dua jenis Autotuning yang dapat dijumpai pada modul control PID komersil: (1) Autotuning metode transient (metode paramterik) dan Autotuning metode umpan balik relay atau metode respon (frekuensi).
- Berdasarkan sifat prosesnya, metode Autotuning respon transient hanya dapat diimplementasikan untuk proses-proses yang bersifat stabil (self regulating proses), sedangkan metode umpan balik relay dapat digunakan baik proses self regulating maun non self regulating



- Karena dalam daerah kerja yang luas, proses umumnya bersifat non linear maka proses autotuning pada dasarnya harus dilakukan pada daerah titik kerja proses
- Untuk proses-proses yang memiliki titik kerja atau perubahan beban yang relatif besar atau juga proses-proses yang memiliki nonlinearity signifikan, maka skema control yang tepat untuk digunakan adalah skema control gain scheduling

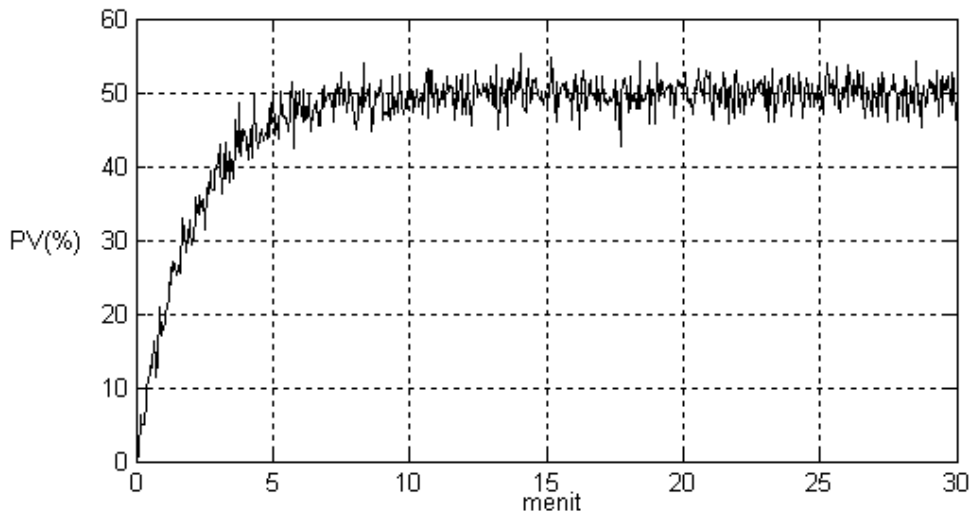
### 3.6 Soal Latihan

**3.1.** Berdasarkan informasi perioda dan magnitude osilasi pada pengontrolan umpan balik relay dalam gambar 3.34 dibawah, carilah parameter PID optimal berdasarkan metode-metode tuning seperti yang tersaji pada tabel 3.2



**Gambar 3.34** Hubungan sinyal kontrol umpan balik relay dengan output proses

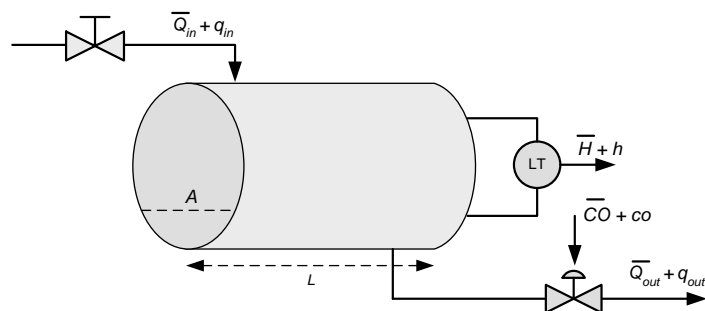
**3.2.** Gambar 3.35 berikut memperlihatkan hasil pengukuran respon output berderau. Carilah lebar histerisis minimal pada sistem kontrol umpan balik relay untuk proses tersebut



**Gambar 3.35** Output proses berderau

**3.3** Carilah fungsi interpolasi penguat proporsional berdasarkan ketinggian pada studi kasus gain scheduling pengontrolan level pada plant nonlinear gambar 3.27.

**3.4** Dengan menggunakan hukum kesetimbangan massa, carilah persamaan dinamika antara level fluida terhadap perubahan sinyal kontrol untuk plant pada gambar 3.36 dibawah



**Gambar 3.36** Plant kontrol level nonlinear

**3.5** Dengan menggunakan simulink, simulasikanlah sistem kontrol umpan balik relay untuk berbagai model proses berikut:

a. 
$$pv(s) = \frac{0.5}{2s+1} e^{-1.5s} .co(s)$$

b. 
$$pv(s) = \frac{-0.5}{2s+1} e^{-1.5s} .co(s)$$

$$\text{c. } pv(s) = \frac{0.5}{2s+1} e^{-1.5s} .co(s) + \frac{1}{2s+1} e^{-0.5s} d(s)$$

$$\text{d. } pv(s) = \frac{-0.5}{s} e^{-1.2s} .co(s)$$

$$\text{e. } pv(s) = \frac{-0.5}{s} e^{-1.2s} .co(s) + \frac{0.1}{s} d(s)$$

$$\text{f. } pv(s) = \frac{0.5}{2s+1} e^{-1.5s} [co(s) + d(s)]$$

$$\text{g. } pv(s) = \frac{-0.5}{s} e^{-1.2s} [co(s) - d(s)]$$

**3.6** Berdasarkan informasi perioda dan magnitude osilasi yang diperoleh dari hasil simulasi, carilah parameter kontrol PI-D untuk proses-proses pada soal 3.5 diatas

ooOoo

## Bab 4

# SKEMA KONTROL PID LANJUT: CASCADE, FEEDFORWARD - FEEDBACK DAN RATIO CONTROL

### Tujuan Bab

Dalam bab ini akan dibahas tiga skema kontrol PID lanjut yang dapat dijumpai di Industri Proses, masing-masing skema *Cascade*, *Feedforward-feedback* dan *Ratio control*. Setelah menyelesaikan bab ini, anda diharapkan dapat

- Menjelaskan prinsip kerja *Cascade control*
- Menjelaskan kelebihan *Cascade control* terhadap kontrol PID loop tunggal
- Menjelaskan syarat-syarat yang harus dipenuhi dalam penerapan *Cascade control*
- Memahami prosedur tuning yang harus dilakukan pada sistem *Cascade control*
- Menjelaskan prinsip kerja Sistem *Feedforward- Feedback Control*
- Menjelaskan kelebihan *Feedforward- Feedback Control* terhadap kontrol PID loop tunggal
- Menjelaskan prinsip kerja *Ratio control*
- Melakukan simulasi untuk masing-masing skema kontrol lanjutan diatas

### Bacaan Prasyarat

- Bab 1. Dinamika dan Model Kontrol Proses Industri
- Bab 2. Kontrol PID Untuk Proses Industri : Berbagai Struktur dan Metoda Tuning Praktis

### 4.1 Pendahuluan

Walaupun hampir semua kebutuhan kontrol proses dapat diselesaikan hanya dengan menggunakan loop PID tunggal, tetapi dalam beberapa kasus tertentu (terutama pengontrolan proses yang memiliki parameter dinamis --delay transportasi dan konstanta waktu proses yang sangat dominan serta proses-proses yang sering mengalami gangguan), pengontrolan dengan loop tunggal umumnya akan menghasilkan unjukkerja yang kurang bahkan tidak memuaskan.

Agar sistem kontrol menghasilkan unjukkerja yang layak, maka pada kasus-kasus diatas diperlukan skema kontrol lanjut yang tidak hanya sekedar loop tunggal melainkan bersifat multiloop.

Tergantung problem yang dihadapi, berikut ini dua topologi kontrol multiloop berbasis algoritma PID yang dapat dijumpai pada pengontrolan proses industri

- Kontrol Bertingkat (*Cascade Control*).

➤ Kontrol Umpan Maju-Umpan Balik (*Feedforward-feedback Control*)

Berdasarkan survey dari *Control Engineering* pada tahun 1998, sekitar 9% dan 6% proses industri berturut-turut menggunakan skema *Cascade control* dan *Feedforward-feedback control* sebagai basis pengontrolannya.

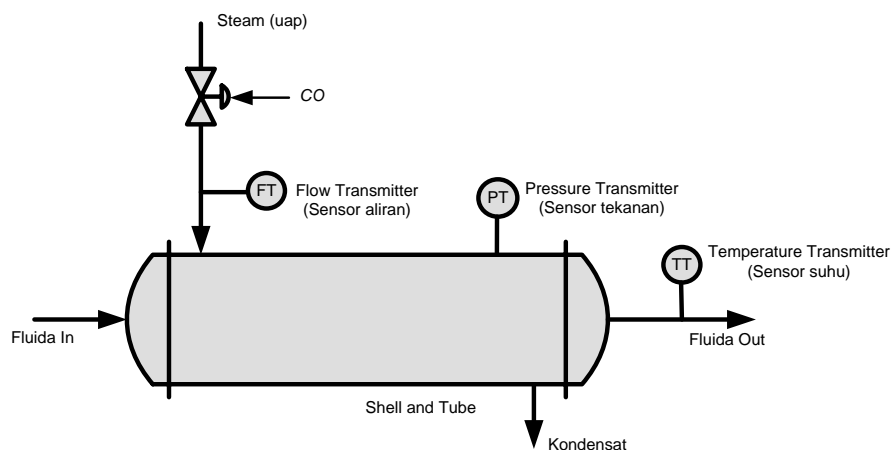
Khusus untuk proses yang melibatkan operasi pencampuran (*mixing*), selain dua skema kontrol diatas, sistem kontrol *multiloop* lain yang dapat dijumpai adalah apa yang dikenal dengan nama *ratio control* (kontrol perbandingan).

#### 4.2 Kontrol Cascade

Kontrol Cascade adalah sebuah metode kontrol yang memiliki (minimal) dua buah loop pengontrolan: loop pengontrolan primer atau *master loop* dan loop pengontrolan sekunder atau *slave loop*. Dalam skema kontrol ini, output kontroler pada sisi master secara fungsional digunakan untuk memanipulasi Setpoint bagi loop pengontrolan sekundernya.

Penerapan skema Cascade pada kontrol proses akan menghasilkan unjuk kerja yang memuaskan jika proses yang dikontrol selama operasinya sering mengalami gangguan pada sisi loop sekundernya.

Salah satu contoh proses industri yang secara intensive memanfaatkan skema cascade dalam pengontrolannya adalah sistem *Heat Exchanger* seperti nampak pada gambar 4.1 berikut.



**Gambar 4.1.** Sistem *Heat Exchanger*

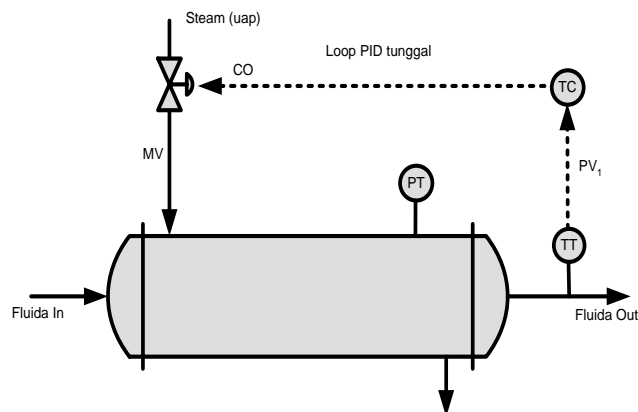
Seperti yang telah dijelaskan dalam **Bab 1**, Hubungan sebab akibat antara perubahan sinyal kontrol penggerak *valve* dengan output (dalam hal ini temperature fluida outlet) untuk proses diatas secara matematis dapat didekati oleh model FOPDT yang hanya dicirikan oleh tiga parameter proses : Keterlambatan transportasi  $-L$ , Konstanta waktu  $-T$  dan Gain statis proses  $-K$ .

Dalam sistem *Heat Exchanger* tersebut, tugas utama yang dilakukan oleh sistem kontrol adalah mengatasi berbagai macam gangguan yang muncul selama operasi pengontrolan berlangsung. Dari sekian banyak sumber gangguan, salah satu yang sering muncul adalah gangguan yang diakibatkan oleh perubahan tekanan *steam* yang digunakan sebagai sumber energi panas.

Perubahan tekanan dapat terjadi karena boiler pemasok *steam* yang sama, umumnya digunakan juga untuk utilitas atau sistem *Heat Exchanger* yang lain: jika salah satu

sistem mulai diaktifkan atau dinonaktifkan, maka tekanan *steam* yang digunakan pada semua plant atau sistem akan mengalami perubahan (gangguan).

Apabila perubahan tekanan *steam* nilainya tidak terlalu besar, skema kontrol PID loop tunggal pada dasarnya telah cukup memadai untuk digunakan sebagai basis pengontrolan (lihat gambar 4.2). Tetapi jika perubahan yang terjadi relative signifikan, pengontrolan loop PID tunggal umumnya akan menghasilkan unjuk kerja yang tidak memuaskan. Hal ini dapat dipahami karena pada sistem kontrol ini, kontroler akan mulai bekerja mengatasi perubahan tekanan hanya ketika telah “melihat” error temperature yang terjadi.



**Gambar 4.2.** Pengontrolan temperature fluida output dengan Loop PID tunggal

Padahal pada satu sisi, karena adanya jarak antara *control valve* dengan plant *Heat Exchanger*, error temperature yang mulai dirasakan pada sisi outlet pada dasarnya disebabkan oleh perubahan tekanan steam yang telah terjadi sekian detik atau menit sebelumnya. Sehingga dengan demikian ketika kontrol PID mulai bekerja untuk mengatasi error, efek perubahan tekanan steam yang telah dulu terjadi masih akan tetap dirasakan pada plant *Heat Exchanger* sekian detik atau menit kedepan. (dengan kata lain kontrol PID akan selalu mengalami keterlambatan aksi kontrol): semakin jauh posisi *control valve* terhadap *plant*, keterlambatan aksi kontrol akan semakin signifikan.

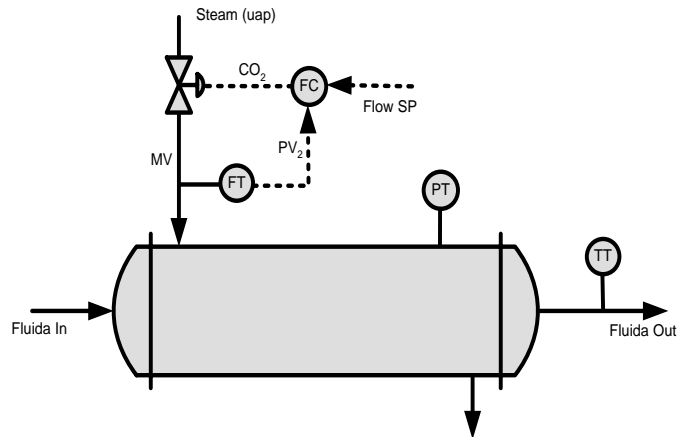
Dengan mengingat:

1. Adanya hubungan kausalitas (ketergantungan) diantara variabel-variabel proses berikut:

Tekanan *steam*  $\longrightarrow$  Laju aliran *steam*  $\longrightarrow$  Temperature fluida outlet

2. Respon perubahan laju aliran *steam* sebagai akibat perubahan tekanan akan jauh lebih cepat dibandingkan respon perubahan temperature fluida sebagai akibat perubahan laju aliran *steam*

Maka untuk mengatasi keterlambatan aksi kontrol pada sistem *Heat Exchanger*, pengontrolan temperature fluida outlet dapat dilakukan dengan meregulasi laju aliran *steam* secara langsung (lihat gambar 4.3).

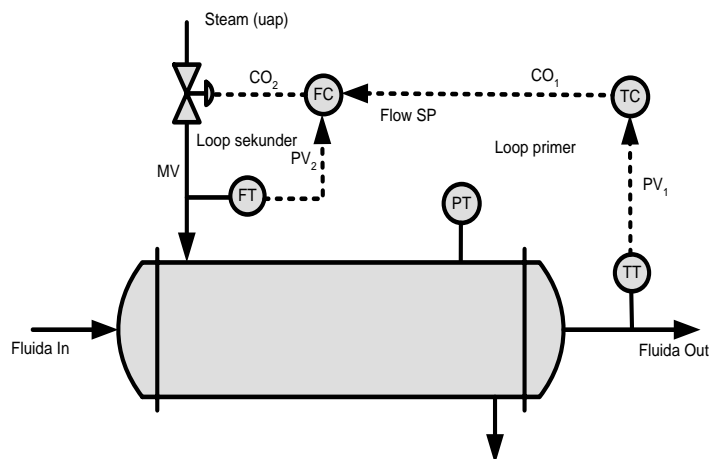


**Gambar 4.3.** Pengontrolan temperature fluida output dapat dilakukan dengan cara meregulasi laju aliran steam

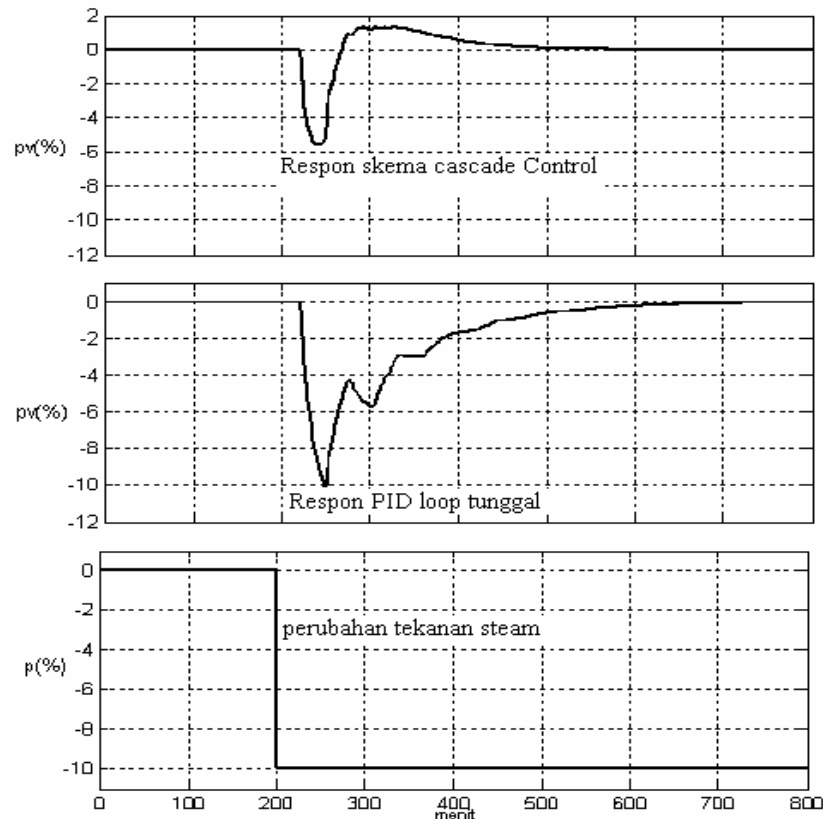
Tetapi ditinjau dari sisi pengontrolan temperature fluida outlet itu sendiri, gambar 4.3 diatas masih bersifat *open loop*: jika terjadi gangguan yang berasal dari sumber lain (misal perubahan laju aliran fluida), maka operator harus menseting ulang setpoint laju aliran *steam* tersebut secara manual.

Agar sistem kontrol temperature pada gambar 4.3 benar-benar bersifat loop tertutup, maka variabel utama yang dikontrol (dalam hal ini temperature fluida outlet) juga perlu diumpkan balik. Tetapi karena pada sistem hanya tersedia satu penggerak *control valve*, maka output kontrol temperature sekarang tidak dapat secara langsung memanipulasi *control valve*, melainkan dijadikan Setpoint bagi laju aliran *steam* (lihat gambar 4.4 ).

Skema pada gambar 4.4 inilah yang dinamakan sebagai kontrol cascade. Dalam hal ini kontrol laju aliran *steam* bertindak sebagai loop pengontrolan sekunder, sedangkan kontrol temperature bertindak sebagai loop pengontrolan primer (kontrol utama).



**Gambar 4.4.** Skema kontrol Cascade pada Sistem *Heat Exchanger*



**Gambar 4.5.** Pengaruh perubahan steam pada sistem kontrol PID tunggal vs sistem kontrol Cascade

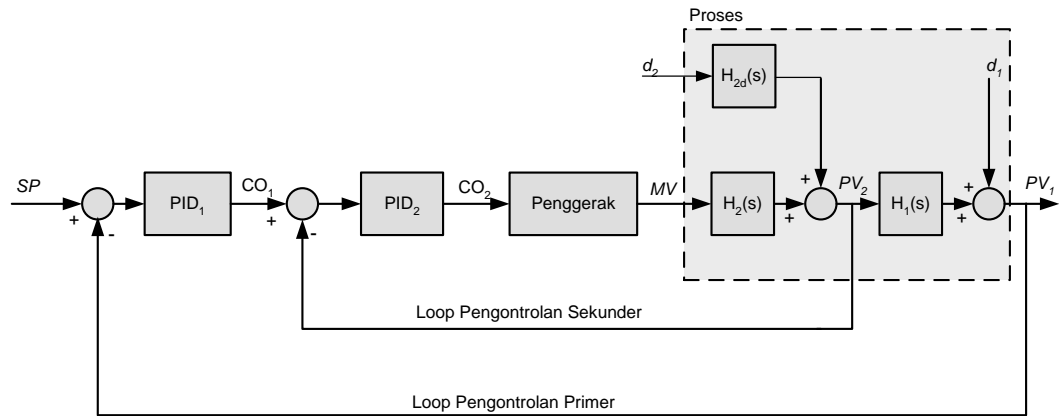
Jika kedua loop kontrol PID pada skema cascade (loop primer dan sekunder) diatas dituning secara tepat, maka unjuk kerja pengontrolan secara praktis akan menunjukkan hasil yang relative lebih baik dibandingkan hasil pengontrolan loop PID tunggal yang dituning secara optimal juga. Sebagai ilustrasi, gambar 4.5 memperlihatkan perbandingan respon perubahan temperature untuk kedua skema kontrol pada sistem *Heat Exchanger* dalam menanggapi perubahan tekanan *steam* berdasarkan hasil simulasi.

#### 4.2.1. Diagram blok dan model matematis proses untuk sistem kontrol Cascade

Relative dibandingkan representasi P&ID-nya, skema *cascade control* ini akan jauh lebih mudah dipahami jika pengontrolan tersebut direpresentasikan dalam bentuk diagram bloknya. Dengan menggunakan diagram blok, hubungan kausalitas diantara variabel-variabel pada sistem *cascade control* dapat terlihat lebih eksplisit (lihat gambar 4.6).

Dengan merujuk pada studi kasus sistem kontrol *Heat Exchanger* sebelumnya,  $pv_2$  dan  $pv_1$  yang tercantum pada gambar 4.6 berturut-turut merepresentasikan perubahan laju aliran *steam* - variabel sekunder dan perubahan temperature fluida outlet - variabel primer (dalam hal ini  $H_2(s)$  adalah fungsi alih perubahan laju aliran *steam* terhadap perubahan tekanan *steam*,  $H_1(s)$  adalah fungsi alih perubahan temperature fluida outlet terhadap perubahan laju aliran *steam*).





**Gambar 4.6.** Blok diagram Struktur Kontrol Cascade

Sedangkan perubahan tekanan *steam* pada sistem *heat exchanger* (gangguan) dalam gambar tersebut dimodelkan sebagai variabel  $d_2$  yang secara langsung mempengaruhi laju aliran *steam*.

Seperti yang telah sedikit disinggung sebelumnya, hubungan antara sinyal kontrol penggerak valve dengan variabel temperature fluida outlet ( $PV_I$ ) secara matematis memiliki model FOPDT:

Dengan menganggap kontrol valve bersifat linear maka  $H_2(s)$  dan  $H_1(s)$  pada gambar 4.6 juga tentunya akan memiliki model matematis linear juga:

$$H_2(s) = \frac{K_2}{T_\gamma s + 1} e^{-L_2 s} \quad (4.1)$$

$$H_1(s) = \frac{K_1}{T_1 s + 1} e^{-L_1 s} \quad (4.2)$$

Tidak hanya pada kasus pengontrolan sistem *heat exchanger*, skema kontrol cascade juga dapat diterapkan pada kasus-kasus pengontrolan proses lainnya (seperti pengontrolan tekanan, level dan sebagainya). Tetapi agar menghasilkan unjuk kerja yang memuaskan, proses yang dikontrol tersebut harus memiliki karakteristik model matematis seperti halnya pada kasus pengontrolan sistem *heat exchanger*, yaitu:

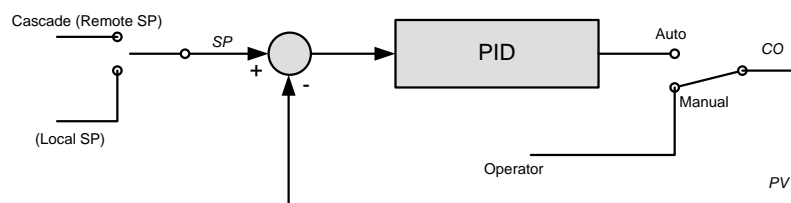
1. Terdapat hubungan erat antara variabel utama yang dikontrol dengan variabel proses lain (variabel sekundernya). Misal dalam sistem *heat exchanger*, jelas terlihat adanya hubungan yang bersifat kausal antara variabel utama yang dikontrol (dalam hal ini temperature) dengan laju aliran *steam*.
2. Gangguan utama yang muncul terdapat pada loop pengontrolan sekunder. Misal pada kasus *heat exchanger*, fungsi utama *cascade control* adalah menekan gangguan yang berasal dari perubahan tekanan steam (loop sekunder). Untuk gangguan yang berasal dari sumber lain (misal perubahan temperature fluida inlet), skema kontrol *cascade* tidak menghasilkan pengaruh yang berarti.
3. Respon loop pengontrolan sekunder harus lebih cepat dibandingkan dengan respon loop pengontrolan primer: konstanta waktu fungsi alih pada loop primer

paling tidak harus sepuluh sampai dua puluh kali lebih besar dari konstanta waktu loop sekundernya.

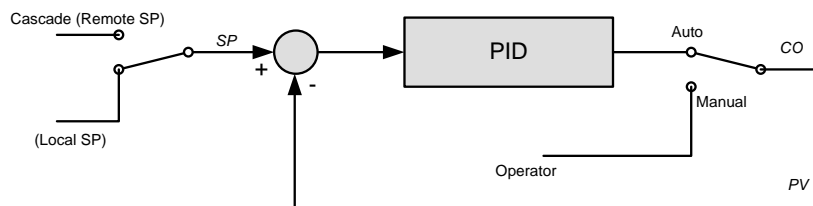
#### 4.2.2 Prosedure Tuning pada Sistem Kontrol Cascade

Secara praktis, skema kontrol cascade umumnya diimplementasikan dalam modul perangkat keras pengontrolan yang sama. Dalam hal ini modul kontrol PID (komersil) yang digunakan selain harus memiliki mode pengontrolan manual dan Auto juga harus memiliki mode kontrol *cascade*.

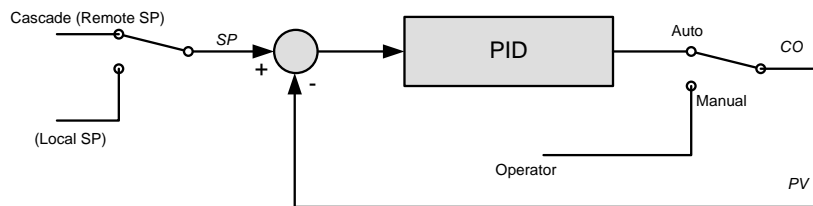
Gambar 4.7 sampai gambar 4.9 berikut memperlihatkan aliran informasi sinyal kontrol untuk masing-masing mode pada modul kontroler.



**Gambar 4.7.** Aliran sinyal kontrol pada mode Manual

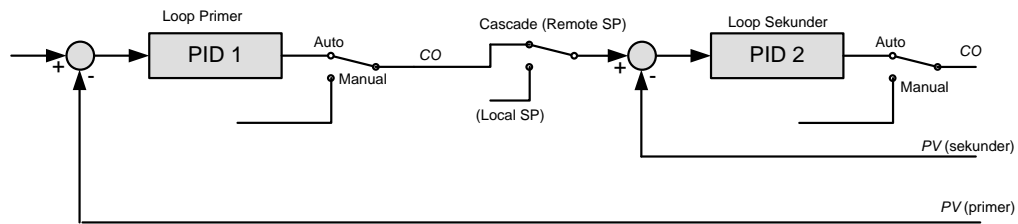


**Gambar 4.8.** Aliran sinyal kontrol pada mode Auto



**Gambar 4.9.** Aliran sinyal kontrol pada mode Cascade

Berdasarkan ketiga gambar diatas nampak bahwa jika kontroler diseting pada mode Manual, maka Output kontroler besarnya diseting secara manual oleh operator (lihat gambar 4.7), jika kontroler diseting pada mode Auto maka output kontroler berasal dari blok kontrol PID, dengan nilai seting point besarnya diatur oleh operator, sedangkan pada mode Cascade besarnya setpoint tidak diatur oleh operator melainkan berasal dari output blok kontrol PID primernya (lihat juga gambar 4.10)



**Gambar 4.10.** Aliran sinyal kontrol lengkap pada mode Cascade

Karena ada dua loop yang saling berinteraksi, tuning sistem kontrol *Cascade* memerlukan perhatian lebih khusus dari sistem kontrol PID loop tunggal. Tuning harus dapat mencapai keadaan loop primer yang stabil dan didukung dengan kestabilan yang mantap dari loop sekunder. Respon kontrol loop sekunder tidak boleh mempengaruhi respon loop primer. Dengan kata lain, loop primer seolah-olah tidak mengetahui bahwa di dalamnya ada loop sekunder. Hal itu dibuat dengan menyetel loop sekunder lebih tidak sensitif dari loop primernya.

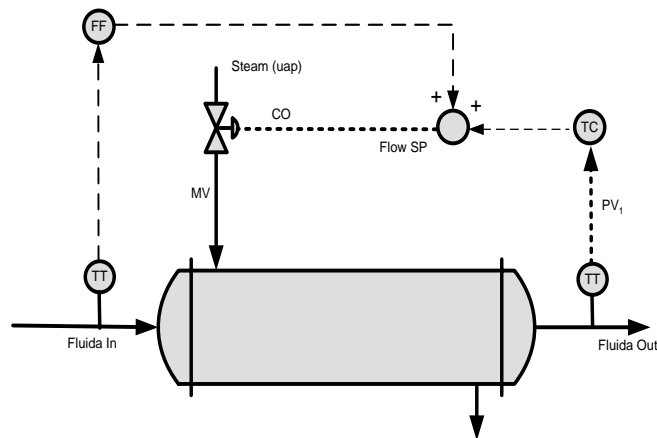
Berikut adalah langkah-langkah untuk men-tuning sistem kontrol Cascade:

1. Meletakkan kedua loop kontrol pada mode manual.
2. Melakukan tuning terlebih dulu pada loop sekunder ( $H_2(s)$ ) seperti halnya tuning sistem kontrol PID loop tunggal (Jika parameter proses  $H_2(s)$  diketahui secara pasti, maka parameter kontrol PID dapat dihitung secara mudah dengan menggunakan metode-metode tuning empiris—lihat kembali bab 2).
3. Setelah setting di loop sekunder menghasilkan respon yang cukup mantap, ubah mode kontrol loop sekunder ke Auto. Kemudian tuning dilakukan pada loop primer, yaitu dengan terlebih dulu mengubah mode loop kontrol sekunder dari posisi Auto ke posisi cascade (Dengan menganggap sistem kontrol memiliki fasilitas *bumpless transfer*, maka secara otomatis, output kontrol primer akan diinisialisasi dengan nilai setpoint dari loop sekundernya).
4. Lakukan tuning loop primer dengan menggunakan metode umpan balik relay atau metode empiris lainnya.

### 4.3.Kontrol Feedforward-Feedback

Sesuai namanya, sistem kontrol *Feedforward-Feedback* adalah sistem kontrol gabungan antara pengontrolan umpan maju (*feedforward*) dan pengontrolan umpan balik (*feedback*). Pada sistem kontrol ini, fungsi utama dari kontrol *feedforward* adalah mengantisipasi pengaruh gangguan terhadap variabel proses yang dikontrol. Dalam hal ini kontroler akan mulai bekerja mengatasi gangguan, sebelum gangguan tersebut mempengaruhi output proses yang dikontrol.

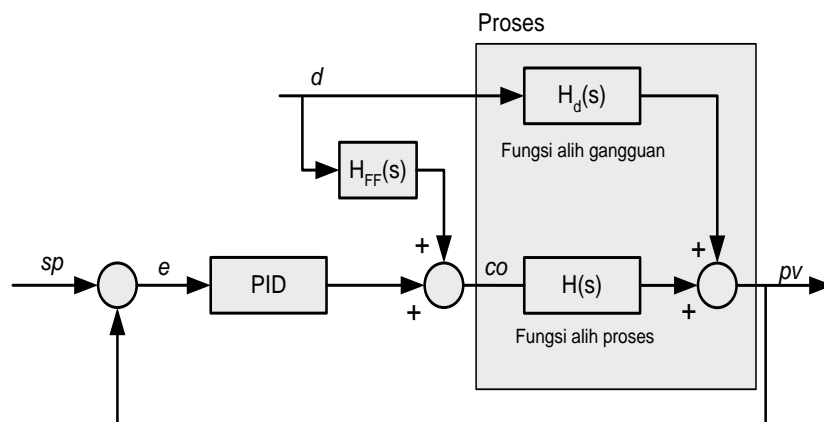
Misal pada sistem kontrol *Heat Exchanger*, skema kontrol *feedforward-feedback* dapat digunakan jika sumber gangguan utama bersalah dari fluktuasi temperature fluida inlet. Pada kasus ini, sistem kontrol *feedforward* akan mulai bereaksi ketika terdeteksi adanya perubahan temperature pada fluida input proses tersebut (bandingkan dengan cara kerja sistem kontrol *feedback*). Dalam bentuk diagram prosesnya, skema kontrol *feedforward-feedback* pada pengontrolan sistem *Heat Exchanger* akan nampak seperti pada gambar 4.11 (dalam diagram tersebut, blok kontrol *feedforward* disimbolkan dengan lingkaran berlabel *FF*).



**Gambar 4.11.** Skema kontrol Feedforward-feedback pada Sistem *Heat Exchanger*

#### 4.3.2. Diagram blok dan model matematis proses untuk sistem kontrol Feedforward-Feedback

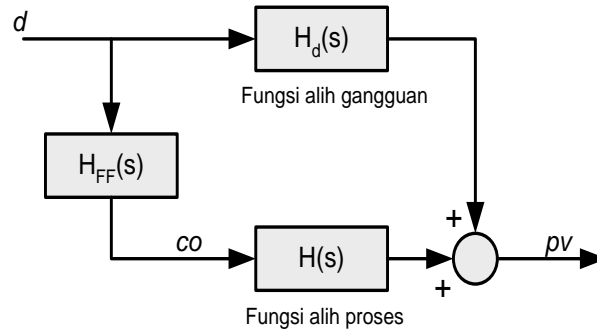
Jika direpresentasikan dalam bentuk diagram bloknya, sistem kontrol *feedforward-feedback* akan nampak seperti pada gambar 4.12 dibawah.



**Gambar 4.12.** Blok diagram struktur kontrol *feedforward-feedback*

Dalam gambar 4.12 diatas,  $d$  adalah gangguan yang dapat diukur (pada kasus sistem kontrol *Heat Exchanger*,  $d$  adalah perubahan temperature fluida inlet), sedangkan  $H_{FF}$  adalah kontrol *Feedforward* yang berfungsi mengkompensasi gangguan tersebut.

Ditinjau secara matematis, blok kontrol  $H_{FF}(s)$  pada dasarnya hanyalah berupa filter atau bahkan gain proporsional saja. Untuk sebuah proses tertentu parameter  $H_{FF}(s)$  dapat dicari dengan memecahkan relasi antara variabel deviasi output proses ( $pv$ ) terhadap pengaruh variabel gangguan ( $d$ ) seperti nampak pada gambar 4.13 dibawah



**Gambar 4.13.** Blok diagram struktur kontrol *feedforward*

Berdasarkan gambar tersebut nampak

$$pv(s) = (H_{FF}(s)H(s) + H_d(s))d(s) \quad (4.3)$$

Agar gangguan  $-d$  yang terjadi tidak mempengaruhi variabel output proses ( $pv$ ), maka fungsi alih proses terhadap gangguan tersebut harus nol, atau secara matematis dapat ditulis:

$$\frac{pv(s)}{d(s)} = H_{FF}(s)H(s) + H_d(s) = 0 \quad (4.4)$$

Dengan menyederhanakan persamaan 4.4, fungsi alih kontrol umpan maju tersebut sekarang dapat dicari secara mudah:

$$H_{FF}(s) = -\frac{H_d(s)}{H(s)} \quad (4.5)$$

Berdasarkan persamaan 4.5 diatas, nampak bahwa fungsi alih kontrol Feedforward dapat dicari jika model proses dan model gangguannya diketahui secara pasti.

Khusus untuk model proses FOPDT:

$$H(s) = \frac{K}{Ts + 1} e^{-Ls}$$

dan

$$H_d(s) = \frac{K_d}{T_d s + 1} e^{-L_d s}$$

Kontrol *Feedforward* dapat dirumuskan menjadi:

$$H_{FF}(s) = -\frac{K_d}{K} \frac{Ts + 1}{T_d s + 1} e^{-(L_d - L)s} \quad (4.6)$$

Dalam sejumlah literature dan buku manual kontrol, persamaan 4.6 diatas umumnya direpresentasikan dalam bentuk berikut:

$$H_{FF}(s) = -K_{FF} \frac{T_{lead} + 1}{T_{lag} s + 1} e^{-L_{FF}s} \quad (4.7)$$

dalam hal ini

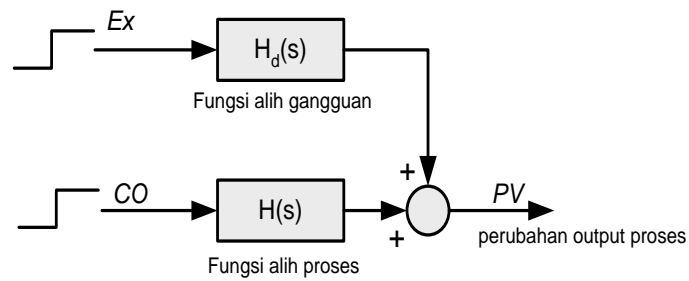
$$K_{FF} = \frac{K_d}{K}$$

$$T_{lead} = T$$

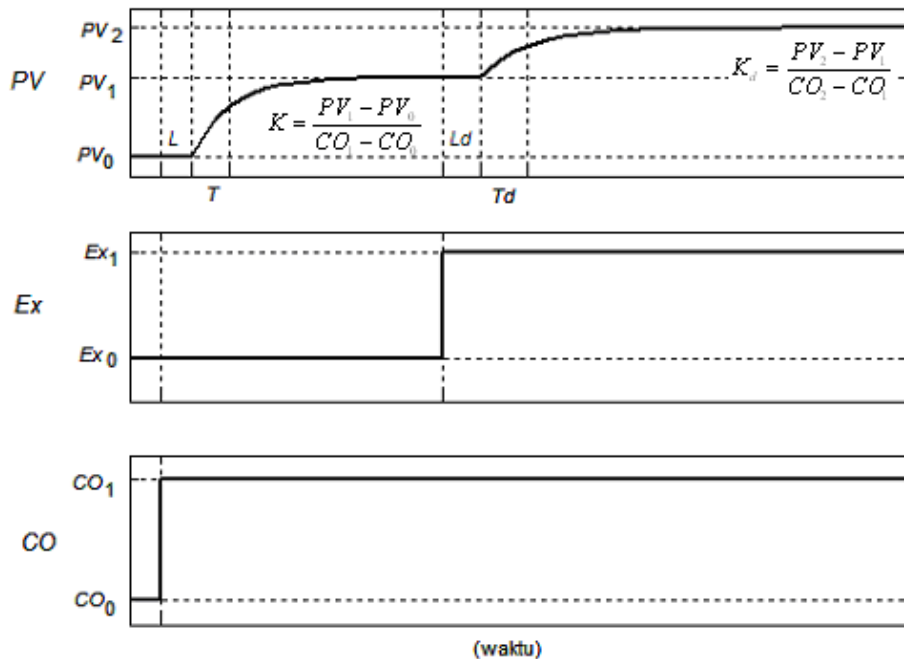
$$T_{lag} = T_d$$

$$L_{FF} = L_d - L$$

Nilai-nilai parameter proses dan gangguan ( $K$ ,  $K_d$ ,  $T$ ,  $T_d$ ,  $L$  dan  $L_d$ ) yang digunakan untuk mendapatkan fungsi alih kontrol *Feedforward* diatas secara praktis dapat dicari lewat eksperimen Bump Test seperti yang diperlihatkan pada gambar 4.14. (tentang mekanisme Bump Test, materinya dapat dilihat kembali pada bab I).



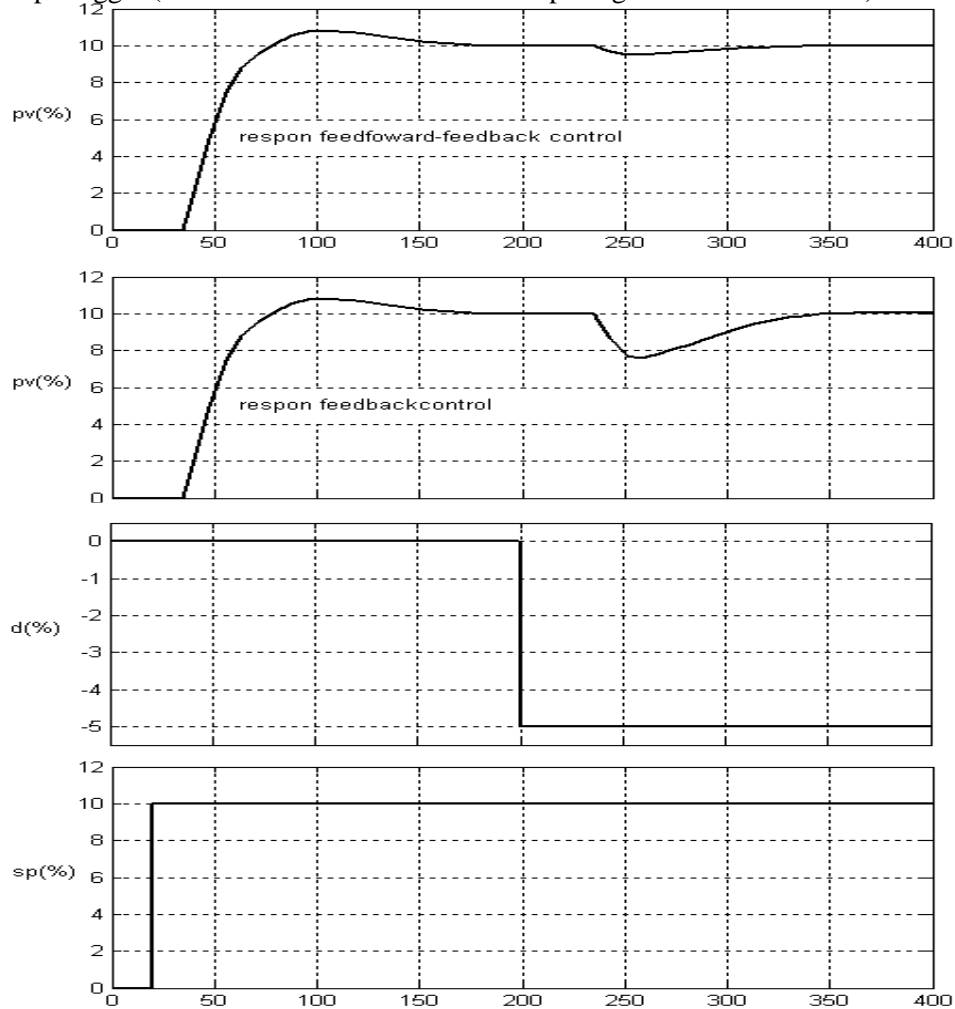
(a). Blok Diagram model proses dan gangguan



(b). Hubungan input-output tipikal hasil eksperimen Bump-test

**Gambar 4.14.** Eksperimen Bump-test untuk mendapatkan parameter model proses dan gangguan

Jika parameter model proses dan model gangguan dapat diketahui secara akurat, maka unjuk kerja pengontrolan dengan menggunakan skema kontrol feedforward-feedback ini akan menunjukkan hasil yang lebih memuaskan dibandingkan dengan pengontrolan PID loop tunggal (lihat ilustrasi hasil dari simulasi pada gambar 4.15 dibawah).

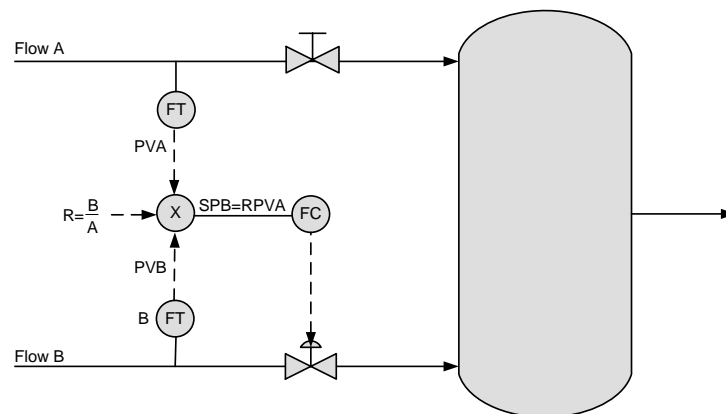


**Gambar 4.15.** Pengaruh gangguan pada sistem kontrol PID tunggal vs sistem kontrol Feedforward – eedback untuk model proses pada persamaan 4.8

#### 4.4. Kontrol Ratio

Tujuan utama kontrol ratio adalah menjaga perbandingan antara dua buah variabel pada nilai yang dikehendaki. Sistem kontrol ini umumnya dapat ditemukan pada proses-proses pencampuran (misal pada sistem pencampuran udara dan bahan bakar pada plant *furnace*)

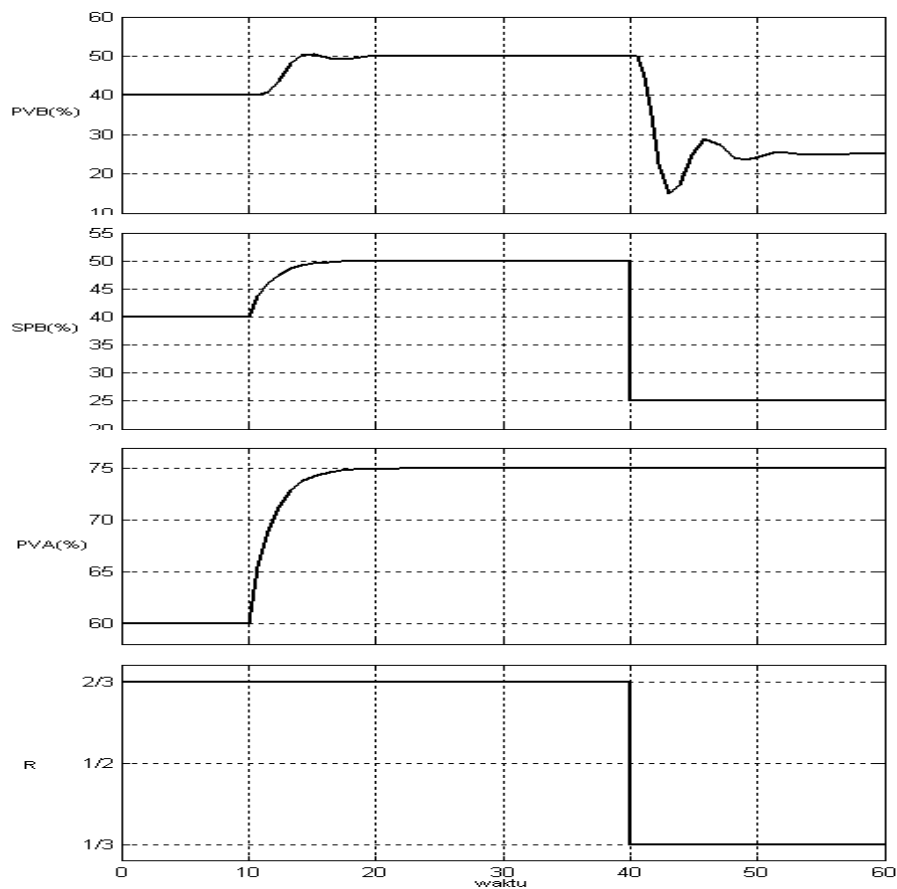
Sebagai contoh, tinjau diagram proses pada gambar 4.16 berikut:



**Gambar 4.16.** Diagram proses pencampuran dua buah fluida

Tujuan kontrol proses diatas adalah menjaga perbandingan antara laju aliran fluida A dengan laju aliran fluida B , atau:

$$R = \frac{PV_B}{PV_A} = \text{konstan}$$



**Gambar 4.16.** Diagram proses pencampuran dua buah fluida



Karena pada diagram proses tersebut, variabel yang dapat dimanipulasi adalah laju aliran B, maka agar perbandingannya tetap konstan, SetPoint untuk laju aliran B tetntunya harus dirumuskan:

$$SP_B = R.PV_A$$

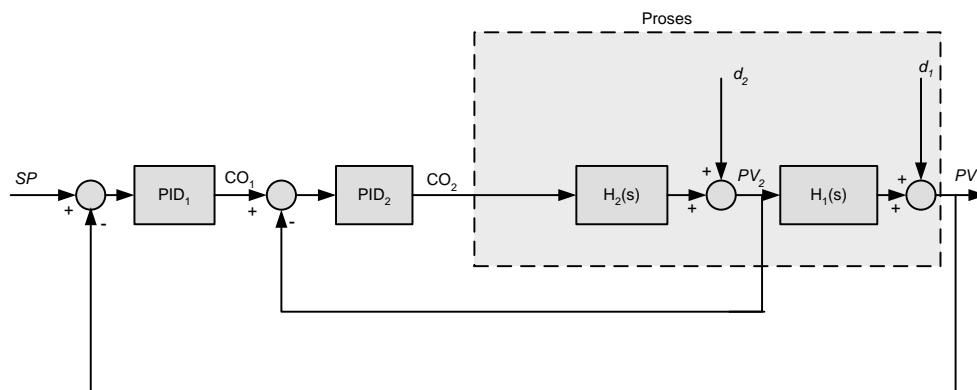
Gambar 4.17 memperlihatkan ilustrasi respon laju aliran B terhadap perubahan laju aliran A (dengan nilai R konstan) dan respon laju aliran B terhadap perubahan R (dengan laju aliran A konstan) pada sebuah sistem kontrol mixing.

## 4.5. Contoh Soal dan Penyelesaian

### Contoh 4.1

#### Perancangan Sistem Kontrol Cascade

Tinjau diagram blok sistem kontrol Cascade pada gambar 4.17 dibawah.



**Gambar 4.17.** Diagram blok sistem kontrol Cascade

Untuk proses tersebut diketahui fungsi alih loop primer ( $H_1(s)$ ) dan loop sekunder ( $H_2(s)$ ) berturut-turut adalah:

$$H_1(s) = \frac{3e^{-6s}}{14s + 1} + d_1(s)$$

dan

$$H_2(s) = \frac{2e^{-s}}{3s + 1} + d_2(s)$$

### Pertanyaan

- Jika kedua kontroler yang digunakan berjenis PI, carilah parameter optimal yang didapat menurut metode Chien Regulator 1
- Dengan menggunakan bantuan Matlab Simulink, simulasikan output sistem kontrol hasil jawab a (terutama terkait dengan terjadinya gangguan  $-d_2$ )

### Penyelesaian

- a. Seperti yang telah dijelaskan pada subbab 4.2., Untuk sistem kontrol cascade, kontroler yang perlu di-tuning terlebih dulu adalah kontroler pada loop sekundernya:

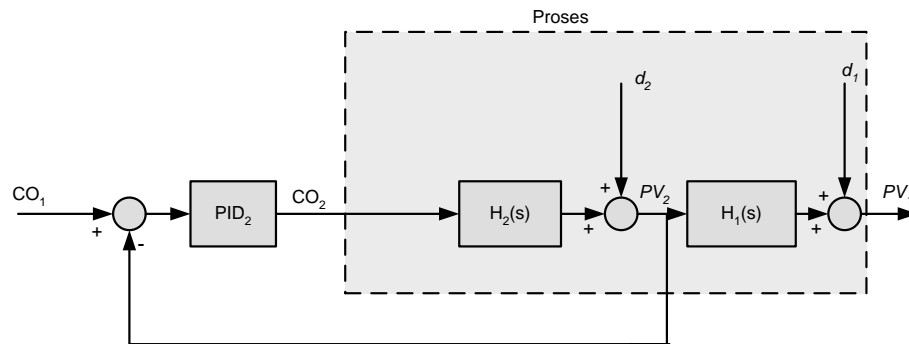
$$H_2(s) = \frac{2e^{-s}}{3s+1} + d_2(s); K=2, T=3, L=1$$

Dengan menggunakan Metode Chien Regulator 1 (lihat tabel 2.4 pada Bab 2) didapat parameter kontrol PI pada loop sekunder tersebut:

$$K_p = \frac{0.6T}{K.L} = \frac{0.6 \times 3}{2 \times 1} = 0.9$$

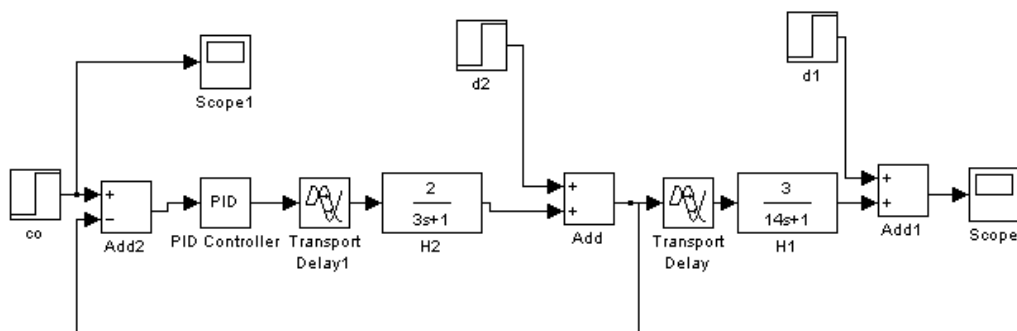
$$T_i = 4L = 4 \times 1 = 4$$

Khusus untuk kontrol PI pada loop primer, parameter kontroler dapat dicari jika terlebih dulu diketahui fungsi alih atau model antara sinyal input loop sekunder  $-CO_1$  terhadap variabel primer yang dikontrol- $PV_1$  (lihat ilustrasinya pada gambar 4.18 dibawah).

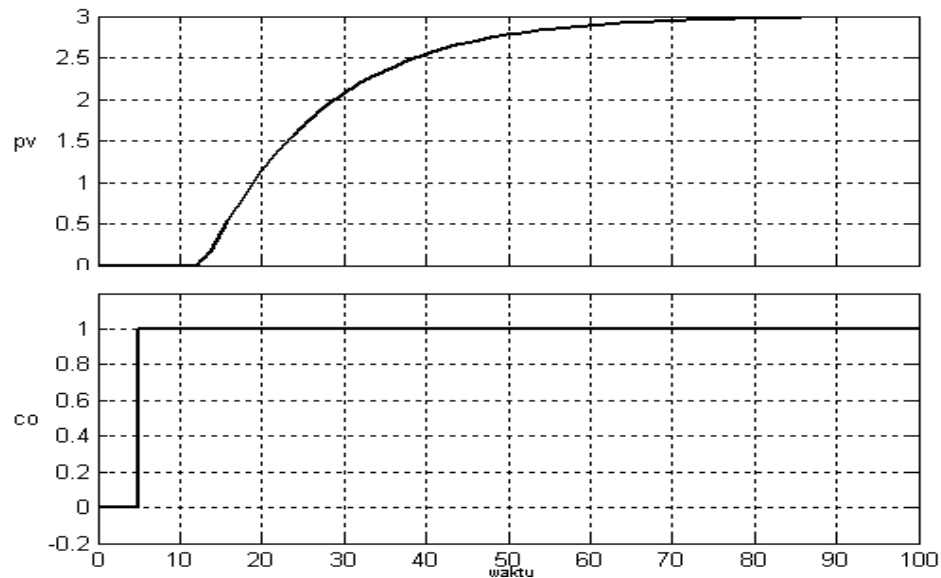


**Gambar 4.18.** Hubungan  $CO_1$  vs  $PV_1$

Dengan menggunakan bantuan Matlab Simulink seperti nampak pada gambar 4.19, hubungan sebab akibat  $CO_1$  vs  $PV_1$  diperoleh pada gambar 4.20



**Gambar 4.19.** Diagram Simulink untuk blok diagram gambar 4.18



**Gambar 4.20.** Hasil simulasi blok simulink gambar 4.19

Berdasarkan identifikasi dari gambar 4.20 diatas, diperoleh parameter proses pada loop primer sistem kontrol tersebut:

Delay transportasi-  $L = 7$ ;

Konstanta proses-  $T = 10$ ;

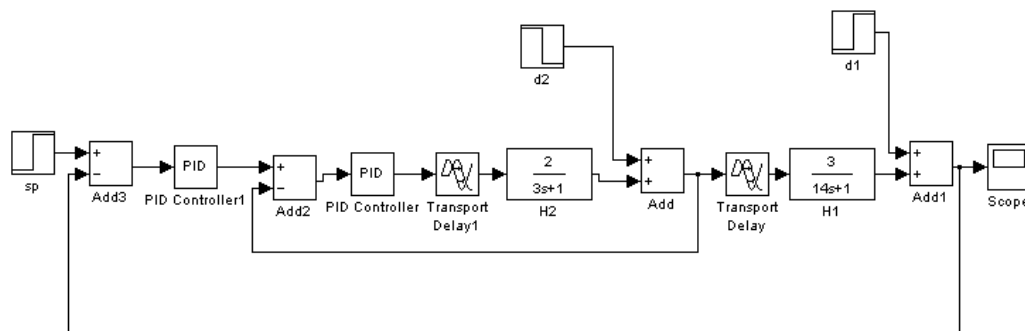
Gain status -  $K=3$

Sehingga dengan demikian, parameter kontrol PI untuk loop primer berdasarkan metode Chien-Regulator I:

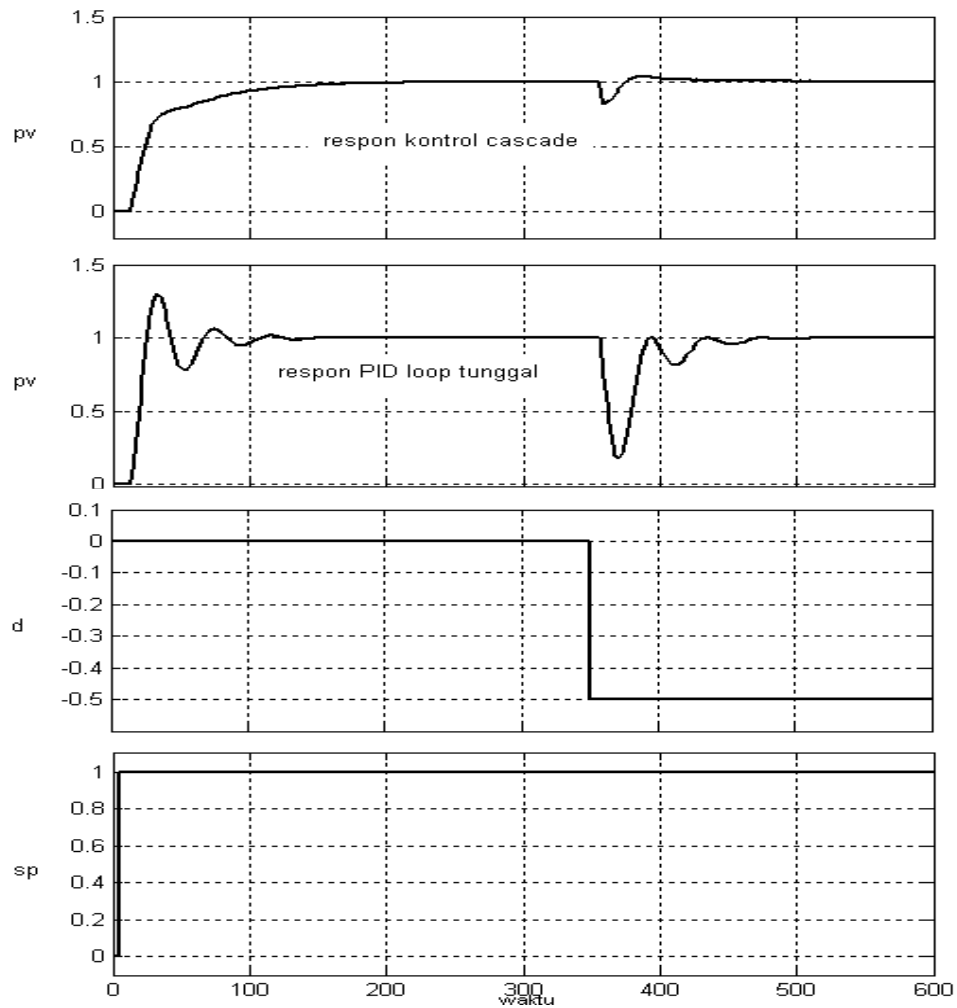
$$K_P = \frac{0.6T}{K.L} = \frac{0.6 \times 10}{3 \times 7} = 0.286$$

$$T_I = 4L = 4 \times 7 = 28$$

- b. Dengan Matlab Simulink (lihat blok simulasi gambar 4.21), hubungan perubahan Setpoint dan input gangguan pada loop sekunder  $-d_2$  terhadap perubahan output proses (variabel primer) dapat dilihat pada gambar 4.22 (sebagai bahan perbandingan pada grafik yang sama di-plot juga hasil simulasi pengontrolan loop PID tunggal pada proses tersebut)



**Gambar 4.21.** Diagram Simulink sistem kontrol Cascade

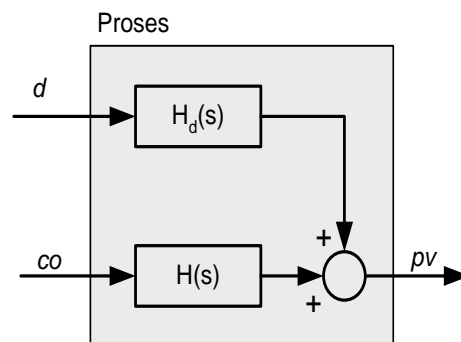


**Gambar 4.22.** Salah satu hasil simulasi blok simulink gambar 4.21

#### Contoh 4.2

#### Perancangan Sistem Kontrol Feedforward-Feedback

Tinjau diagram blok proses pada gambar 4.23 berikut



**Gambar 4.23.** Diagram blok proses dengan gangguan

Berdasarkan hasil identifikasi Bump Test, proses tersebut memiliki model matematis :

$$pv(s) = \frac{0.8e^{-5s}}{12s+1}co(s) - \frac{1e^{-7}}{15s+1}d(s)$$

### Pertanyaan

- Jika kontroler feedback yang digunakan pada proses tersebut adalah PID ideal, carilah parameter optimal yang didapat menurut metode Chien Regulator 0% overshoot
- Rancang sistem kontrol Feedforward untuk proses diatas
- Dengan menggunakan Matlab Simulink, simulasikan tanggapan output proses terhadap perubahan Setpoint dan gangguan, masing-masing untuk kontrol PID loop tunggal (hasil jawab a) dan kontrol Feedforward-feedback (hasil jawab a dan b).

### Penyelesaian

- Berdasarkan persamaan model matematis proses tersebut, diketahui :

Gain statis proses –  $K = 0.8$

Konstanta waktu proses –  $T = 12$

Delay transportasi proses –  $L = 5$

Dengan demikian Parameter PID menurut metode Chien-Regulator 0% overshoot (lihat tabel 2.7 pada bab 2).

$$K_p = \frac{0.95T}{KL} = \frac{0.95 \times 12}{0.8 \times 5} = 2.85$$

$$T_i = 2.38L = 2.38 \times 5 = 11.9$$

$$T_D = 0.42L = 0.42 \times 5 = 2.1$$

- Untuk merancang sistem kontrol feedforward, perlu diketahui terlebih dulu model gangguan proses tersebut, berdasarkan soal didapat:

Gain statis gangguan –  $K_d = -1$

Konstanta waktu gangguan –  $T_d = 15$

Delay transportasi gangguan –  $L_d = 7$

Dengan merujuk persamaan 4.7, kontrol Feedforward dirumuskan:

$$H_{FF}(s) = -K_{FF} \frac{T_{lead} + 1}{T_{lag}s + 1} e^{-L_{FF}s}$$

Dalam hal ini:

$$K_{FF} = \frac{K_d}{K} = \frac{-1}{0.8} = 1.25$$

$$T_{lead} = T = 12$$

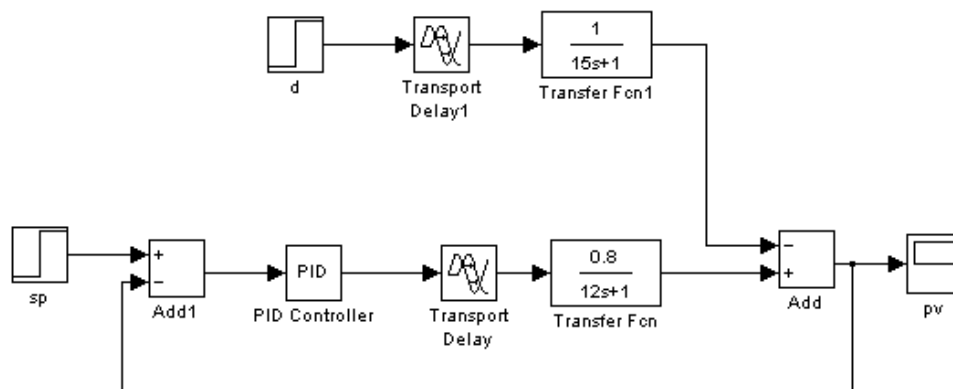
$$T_{lag} = T_d = 15$$

$$L_{FF} = L_d - L = 7 - 5 = 2$$

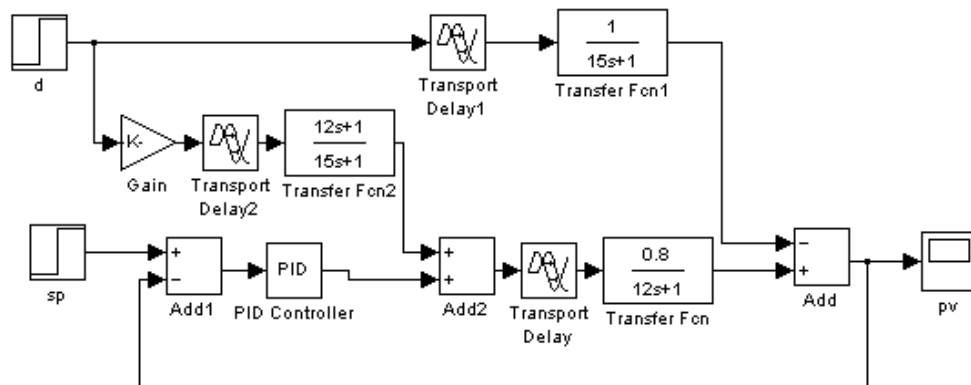
Sehingga dengan demikian:

$$H_{FF}(s) = 1.25 \frac{12+1}{15s+1} e^{-2s}$$

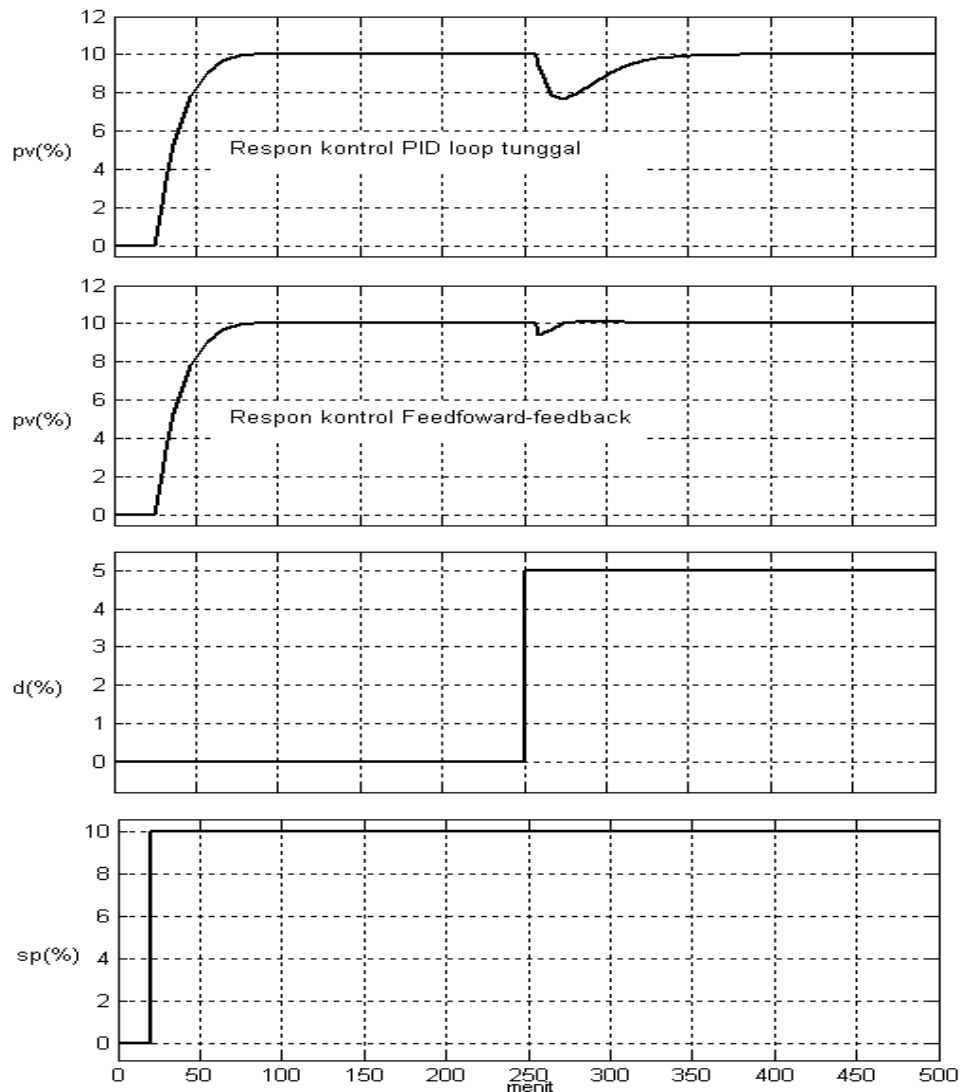
- c. Dengan Matlab Simulink (lihat blok simulasi gambar 4.24 dan gambar 4.25), hubungan perubahan Setpoint dan input gangguan terhadap perubahan output proses dapat dilihat pada gambar 4.26



**Gambar 4.24.** Diagram Simulink sistem kontrol Loop PID tunggal untuk contoh soal 4.2.c



**Gambar 4.25.** Diagram Simulink sistem kontrol Feedfoward-feedback untuk contoh soal 4.2.c

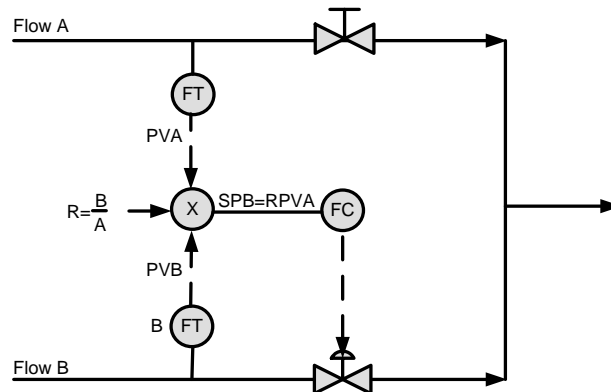


**Gambar 4.26.** Diagram blok sistem kontrol Cascade

### Contoh 4.3

#### Perancangan Sistem Kontrol Ratio

Tinjau proses *mixing* dua jenis fluida pada gambar 4.27 dibawah.  
Dalam proses pencampuran tersebut, hanya fluida B yang dikontrol



**Gambar 4.27.** Diagram blok sistem kontrol proses *mixing*

### Pertanyaan

- a. Jika diketahui model matematis antara sinyal kontrol terhadap laju fluida B:

$$pv(s) = \frac{0.48e^{-5s}}{20s + 1} co(s)$$

Carilah parameter PID ideal berdasarkan metode Chien-Regulator 0% overshoot untuk sistem kontrol tersebut

- b. Dengan menggunakan Matlab Simulink, simulasikan tanggapan output proses laju aliran B terhadap perubahan laju aliran A (dengan nilai R konstan) dan respon laju aliran B terhadap perubahan R (dengan laju aliran A konstan) pada sistem tersebut

### Penyelesaian

- a. Berdasarkan persamaan model matematis proses diatas, didapat parameter kontrol PID ideal:

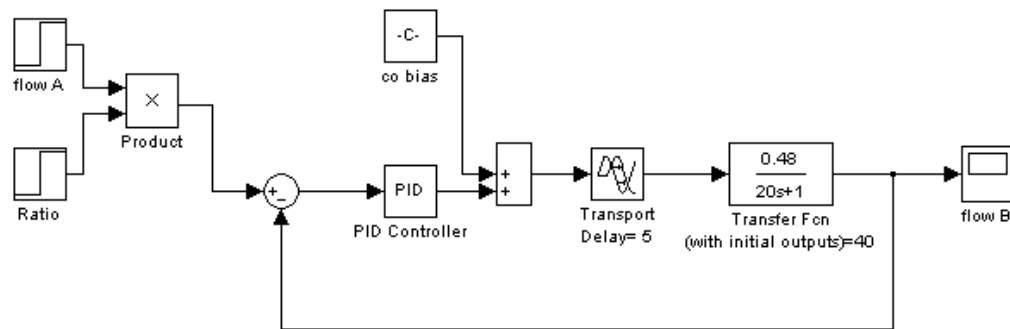
$$K_P = \frac{0.95T}{KL} = \frac{0.95 \times 20}{0.48 \times 5} = 7.9$$

$$T_I = 2.38L = 2.38 \times 5 = 11.9$$

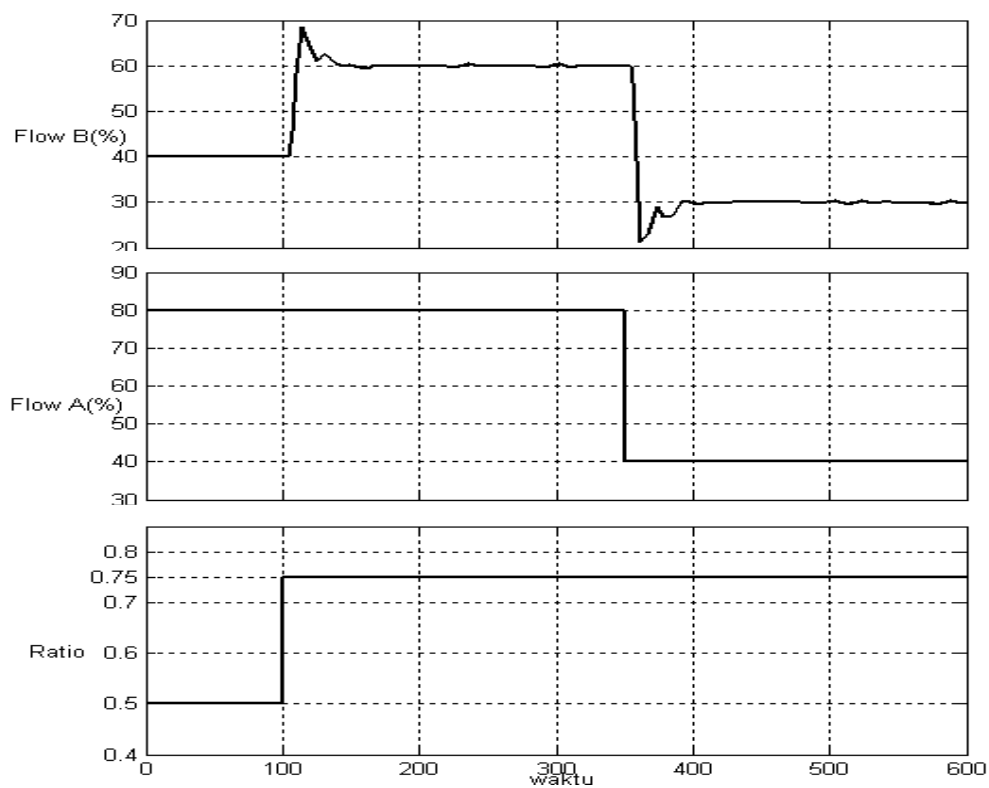
$$T_D = 0.42L = 0.42 \times 5 = 2.1$$

- b. Dengan Matlab Simulink (lihat blok simulasi gambar 4.28), tanggapan perubahan laju aliran B terhadap perubahan laju aliran A (dengan nilai R konstan) dan respon laju aliran B terhadap perubahan R (dengan laju aliran A konstan) pada sistem kontrol tersebut nampak pada gambar 4.29.





**Gambar 4.28.** Diagram Simulink sistem kontrol Feedforward-feedback untuk gambar 4.26



**Gambar 4.29.** Salah satu hasil simulasi blok Simulink gambar 4.28

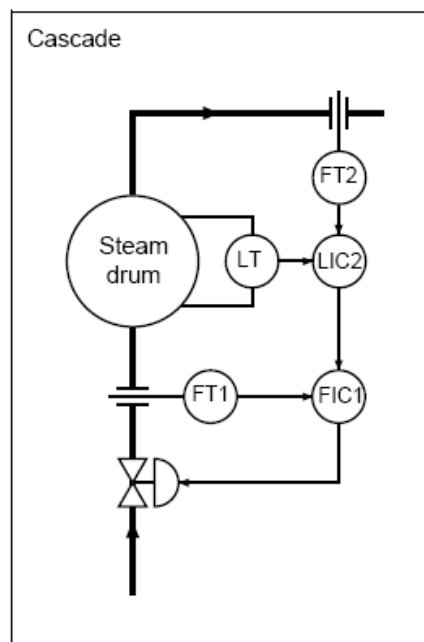
#### 4.6 Ringkasan

- Sistem kontrol Cascade adalah sistem kontrol yang memiliki minimal dua buah loop pengontrolan: Loop pengontrolan primer dan loop pengontrolan sekunder
- Skema kontrol cascade hanya efektif digunakan jika memenuhi dua syarat utama berikut:
  - Sumber gangguan langsung mempengaruhi loop pengontrolan sekunder
  - Dinamika loop sekunder jauh lebih cepat dibandingkan loop primernya

- Skema kontrol Feedforward-feedback adalah sistem kontrol gabungan antara kontrol umpan maju dan sistem kontrol umpan balik
- Dalam sistem kontrol Feedforward-feedback, elemen kontrol feedforward digunakan untuk mengantisipasi gangguan yang muncul pada sisi input proses yang dikontrol. Syarat penerapannya adalah gangguan tersebut dapat diukur secara langsung.
- Seperti tersirat dari namanya, kontrol Ratio adalah sistem kontrol yang berfungsi meregulasi atau menjaga perbandingan dua atau lebih variabel proses. Sistem kontrol ini banyak dijumpai pada proses *mixing*
- Secara teknis, sistem kontrol Ratio pada dasarnya merupakan bentuk sederhana dari sistem feedforward, dimana letak kontroler ada di luar loop umpan balik utama.

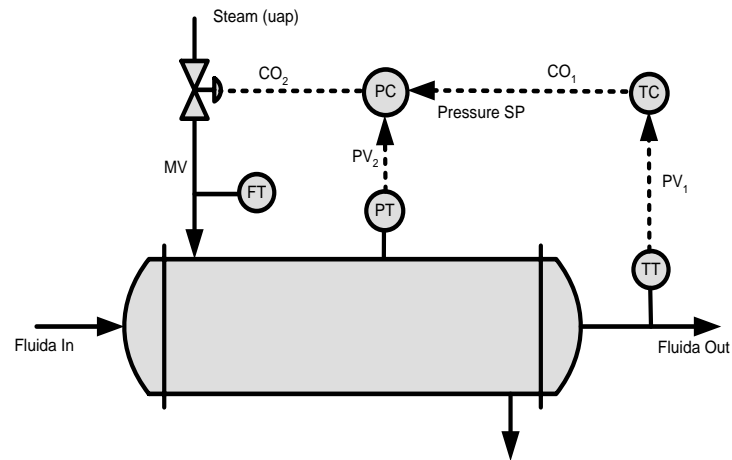
#### 4.7 Soal Latihan

- 4.1. Analisis dan gambarkan diagram blok untuk sistem kontrol Cascade pada gambar 4.30 berikut



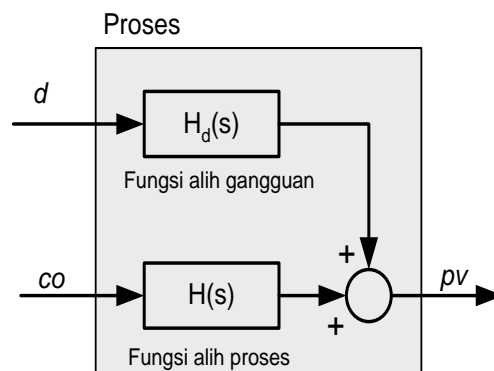
**Gambar 4.30.** Diagram blok sistem kontrol Cascade untuk soal 4.1

- 4.2. Analisis dan gambarkan diagram blok untuk sistem kontrol Cascade pada gambar 4.31 berikut



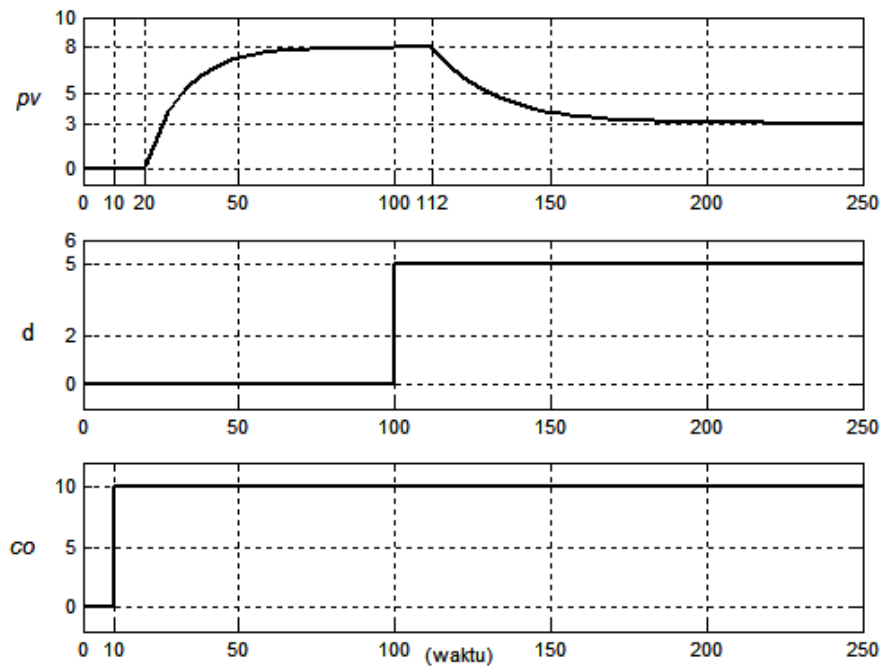
**Gambar 4.31.** Diagram blok sistem kontrol Cascade untuk soal 4.2

4.3. Tinjau diagram blok proses pada gambar 4.32 berikut



**Gambar 4.32.** Diagram blok proses dengan gangguan

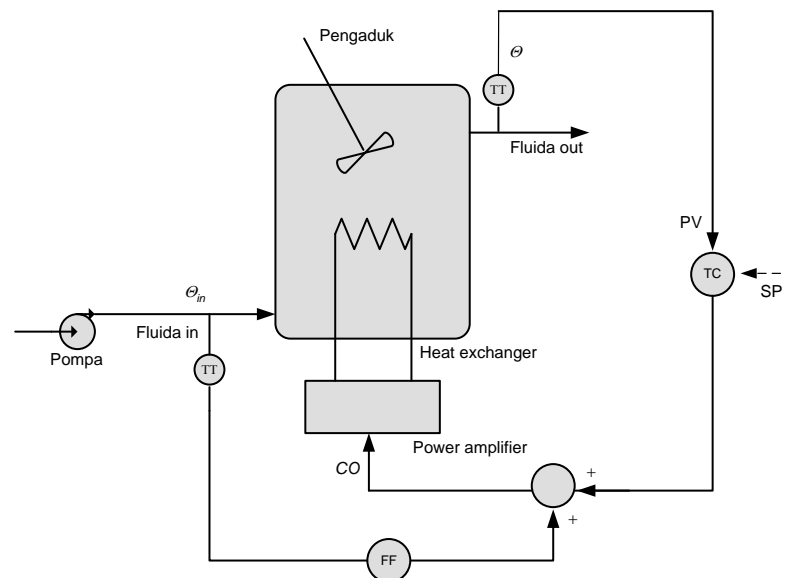
Berdasarkan hasil identifikasi Bump Test, diperoleh grafik hubungan  $co$  dan  $d$  terhadap  $pv$  untuk proses tersebut seperti nampak pada gambar 4.33



**Gambar 4.33.** grafik hubungan co dan d terhadap pv

**4.4.** Berdasarkan hasil penyelesaian **contoh soal 1.8 dalam Bab 1**, dinamika temperature fluida output pada sistem *Heat Exchanger* gambar 3.34 dibawah adalah:

$$\Theta(s) = \frac{Ke^{-Ls}}{Ts + 1} CO(s) + \frac{e^{-Ls}}{Ts + 1} \Theta_{in}(s)$$



**Gambar 4.34.** Diagram proses kontrol Feedforward-feedback pada sistem Heat -Exchanger

dalam hal ini

$$\text{Konstanta waktu proses: } T = \frac{V\rho}{w}$$

$$\text{Gain statis proses: } K = \frac{K_{ps}}{cw}$$

$$\text{Delay transportasi: } L = \frac{K_L}{w}$$

Jika parameter proses diatas berturut-turut memiliki nilai nominal

$$V \text{ (Volume tangki)} = 0.2 \text{ m}^3$$

$$\rho \text{ (densitas fluida)} = 1000 \text{ kg/m}^3$$

$$c \text{ (Kalor jenis fluida)} = 4000 \text{ J/(Kg}^\circ\text{C)}$$

$$K_{ps} \text{ (konstanta kesebandingan antara perubahan panas dengan perubahan sinyal kontrol)} = 800 \text{ W/\% (800.60 J/menit/\% = 48000 J/menit/\%)}$$

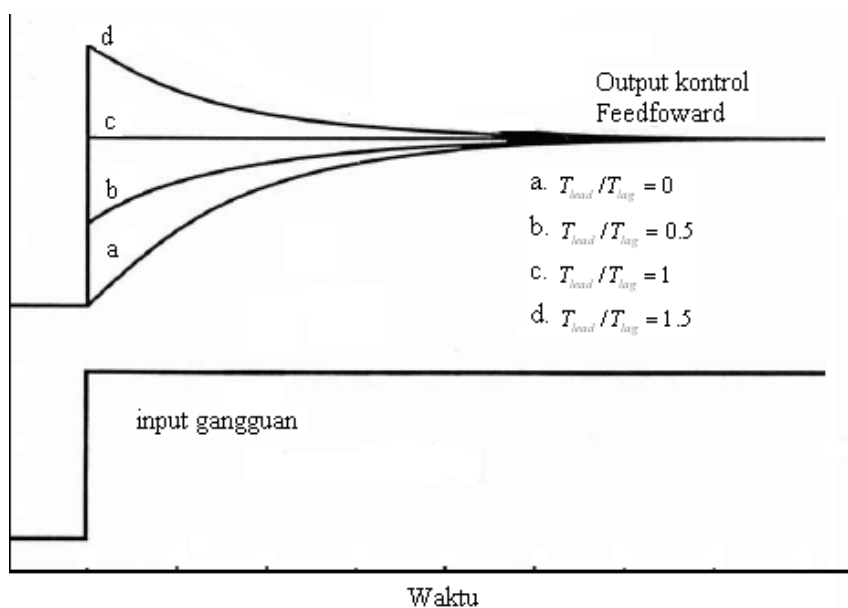
$$K_L = 50 \text{ (Kg)}$$

$$w \text{ (laju aliran fluida)} = 20 \text{ kg/menit}$$

#### Pertanyaan

- Jika kontroler feedback yang digunakan pada proses tersebut adalah PID ideal, carilah parameter optimal yang didapat menurut metode Ziegler-Nichols
- Rancang sistem kontrol Feedforward untuk proses diatas
- Dengan menggunakan Matlab Simulink, simulasikan tanggapan output proses terhadap perubahan Setpoint dan gangguan, masing-masing untuk kontrol PID loop tunggal (hasil jawab a) dan kontrol Feedforward-feedback (hasil jawab a dan b).

**4.5.** Untuk perbandingan  $T_{lead}/T_{lag}$  atau  $T/T_d$  tertentu, output kontrol  $H_{FF}(s)$  akan nampak seperti pada gambar 4.35 dibawah



**Gambar 4.35.** Hubungan tipikal input-output pada blok kontrol  $H_{FF}(s)$

---

Dengan menggunakan bantuan matlab Simulink, buktikan grafik-grafik yang nampak pada gambar 4.35 tersebut.

4.5. Rancang dan simulasikan sistem kontrol cascade untuk pengontrolan dua buah proses berikut:

a. Fungsi alih loop primer:

$$H_1(s) = \frac{0.8e^{-10s}}{16s+1} + d_1(s)$$

Fungsi alih loop sekunder:

$$H_2(s) = \frac{2e^{-2s}}{5s+1} + d_2(s)$$

b. Fungsi alih loop primer:

$$H_1(s) = \frac{0.8e^{-s}}{4s+1} + d_1(s)$$

Fungsi alih loop sekunder:

$$H_2(s) = \frac{3e^{-12s}}{15s+1} + d_2(s)$$

4.6. Berdasarkan hasil simulasi dari jawaban 4.5. buat kesimpulan tentang unjuk kerja sistem kontrol cascade terkait dengan dinamika proses loop primer vs loop sekunder.

4.7. Rancang dan simulasikan sistem kontrol feedforward-feedback untuk pengontrolan tiga buah proses berikut:

a. 
$$pv(s) = \frac{0.5e^{-5s}}{10s+1}co(s) - \frac{1e^{-10}}{20s+1}d(s)$$

b. 
$$pv(s) = \frac{0.5e^{-5s}}{10s+1}co(s) - \frac{1e^{-5}}{10s+1}d(s)$$

c. 
$$pv(s) = \frac{0.5e^{-10s}}{20s+1}co(s) - \frac{1e^{-5}}{10s+1}d(s)$$

4.8. Berdasarkan hasil simulasi dari jawaban 4.7. buat kesimpulan tentang unjuk kerja sistem kontrol cascade terkait dengan dinamika proses loop primer vs loop sekunder.

ooOoo

---

## BAB 5

# KONTROL PID DIGITAL DAN REALISASI PROGRAM

### Tujuan Bab

Khusus dalam bab ini akan dibahas PID bentuk diskret yang realisasi praktisnya dapat dijumpai pada modul PID digital. Setelah menyelesaikan bab ini, diharapkan anda dapat:

- Menjelaskan perbedaan kontrol PID analog dan kontrol PID digital
- Menjelaskan prinsip kerja sistem kontrol PID digital
- Melakukan diskritisasi PID analog kedalam versi PID digitalnya
- Mengimplementasikan kontrol PID digital kedalam bentuk algoritma komputer
- Merealisasikan *bumpless transfer* dan *integrator windup* dalam bentuk program

### Bacaan Prasyarat

- Bab 2. Kontrol PID Untuk Proses Industri : Berbagai Struktur dan Metoda Tuning Praktis

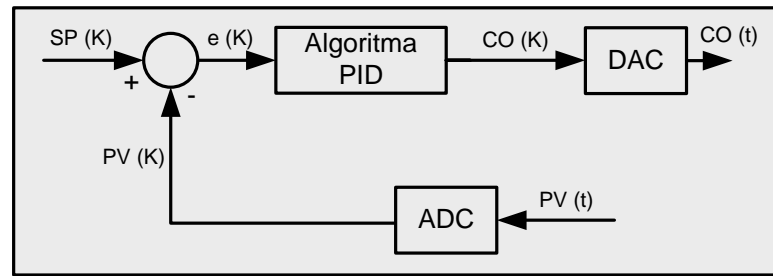
### 5.1 Pendahuluan

Istilah PID digital pada dasarnya mengacu pada jenis perangkat keras digital dimana sistem kontrol PID tersebut ditanamkan. Berbeda dengan kontrol PID analog yang realisasi praktisnya dijumpai dalam bentuk perangkat keras rangkaian elektronika analog, sistem kontrol PID digital implementasinya dapat dijumpai dalam bentuk persamaan matematis yang ditanam pada sistem microprocessor.

Dewasa ini hampir dapat dipastikan modul kontrol PID yang terinstal di industri atau modul komersil yang beredar di pasaran telah didominasi oleh modul digital dengan basis sistem *microprocessor*. Relative dibandingkan modul analog, modul PID digital memiliki beberapa kelebihan, diantaranya yaitu:

- Dapat diintegrasikan secara mudah dengan sistem lain membentuk sebuah jaringan kontrol
- Banyak fungsi dan fitur tambahan yang tidak dapat ditemukan dalam modul PID analog
- Kepresisian sinyal kontrol PID digital tidak tergantung komponen yang digunakan

Dalam bentuk diagram blok, kontrol PID digital dapat diilustrasikan seperti nampak pada gambar 5.1. Karena proses yang dikontrol bersifat kontinyu atau analog, maka di dalam modul ini diperlukan perangkat keras tambahan berupa ADC (*Analog to Digital Converter*) dan DAC (*Digital to Analog Converter*) yang digunakan sebagai antara muka kontroler digital dengan proses.



**Gambar 5.1.** Diagram blok Modul PID digital

Berbeda dengan kontrol PID analog yang pengolahannya bersifat kontinu, Di dalam sistem *microprocessor*, pengolahan sinyal kontrol oleh modul digital dilakukan hanya pada waktu-waktu diskret. Dalam hal ini, konversi sinyal dari analog ke digital, pengolahan sinyal error, sampai konversi balik digital ke analog dilakukan pada interval atau waktu cuplik (sampling) –  $T_c$  tertentu.

Dalam modul kontrol PID komersil, waktu cuplik tersebut umumnya dapat dipilih secara bebas. Pemilihan waktu cuplik pada dasarnya sangat tergantung dari proses yang dikontrol: Semakin cepat dinamika sebuah proses, waktu cuplik yang dipilih atau yang digunakan juga harus semakin cepat. Tabel 5.1 berikut memperlihatkan waktu cuplik yang umum digunakan pada modul digital dalam pengontrolan beberapa jenis variabel proses

**Tabel 5.1.** Lebar waktu cuplik yang umum digunakan untuk beberapa jenis variabel proses

No	Jenis Variabel	Waktu Cuplik - $T_c$ (detik)
1	Aliran (flow)	1-3
2	Level	5-10
3	Tekanan	1-5
4	Temperature	10-20

## 5.2 Persamaan PID Digital dan Realisasi Program

Seperti telah sedikit disinggung sebelumnya, realisasi kontrol PID digital akan ditemukan dalam sistem berbasis *microprocessor*. Karena sistem *microprocessor* adalah sebuah sistem pengolah data yang bekerja pada basis-basis waktu diskret, maka kontrol PID yang ditanam didalamnya pun harus memiliki persamaan matematis diskret (dalam hal ini, eksekusi kontrol PID hanya dilakukan pada waktu-waktu cuplik saja).

Salah satu langkah termudah untuk mendapatkan versi diskret dari kontrol PID digital adalah dengan cara diskritisasi persamaan PID analog asosiasinya. Ketelitian PID digital yang didapat dari diskritisasi ini sangat tergantung dari lebar waktu cuplik yang digunakan: Semakin cepat waktu cuplik, maka perilaku PID digital tersebut akan semakin sama dengan PID analog asosiasinya.

Sebagai studi kasus, khusus dalam subbab ini akan ditinjau diskritisasi persamaan PID analog jenis ideal seperti nampak pada persamaan 5.1 berikut:

$$CO(t) = K_p \left[ e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right] \quad (5.1)$$



Untuk kemudahan mendapatkan versi diskretnya, persamaan diatas kita susun ulang seperti nampak pada persamaan 5.2 (lihat juga diagram bloknya pada gambar 5.2)

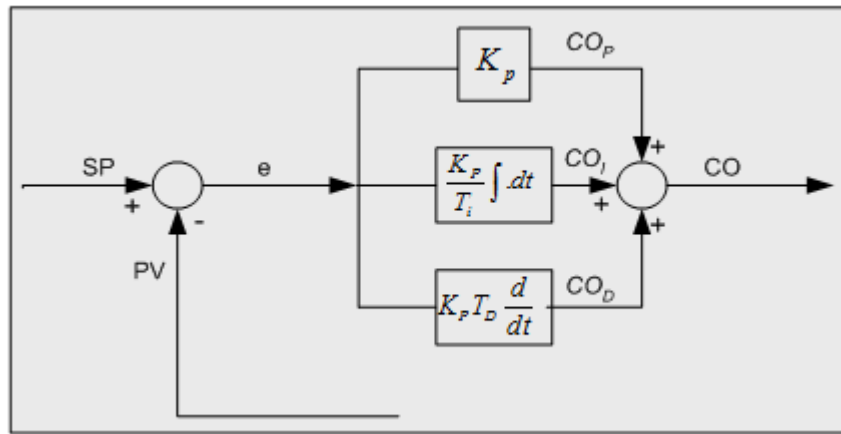
$$CO(t) = CO_p(t) + CO_I(t) + CO_D(t) \quad (5.2)$$

dalam hal ini,

$$CO_p(t) = K_p e(t) \quad (5.3)$$

$$CO_I(t) = \frac{K_p}{T_i} \int_0^t e(\tau) d\tau \quad (5.4)$$

$$CO_D(t) = K_p T_D \frac{de(t)}{dt} \quad (5.5)$$



**Gambar 5.2.** Diagram blok Modul PID digital

Dengan mengacu pada persamaan 5.2, PID ideal versi diskret-nya sekarang dapat diperoleh dengan terlebih dulu mencari versi diskret untuk masing-masing suku persamaan penyusun kontrol PID ideal tersebut (persamaan 5.3-persamaan 5.5).

Khusus untuk output suku kontrol Proporsional (persamaan 5.3), karena bentuk persamaannya statis, versi diskret nya dapat diperoleh langsung tanpa menggunakan pendekatan apapun, hal ini seperti nampak pada persamaan 5.6 berikut:

$$CO_p(k) = K_p e(k) \quad (5.6)$$

Berdasarkan persamaan 5.6 diatas, terlihat bahwa output kontrol Proporsional untuk waktu cuplik tertentu (waktu cuplik ke -k) pada dasarnya hanya tergantung pada error saat pencuplikan itu juga.

Untuk suku output kontrol Integral (persamaan 5.4) dan output kontrol Diferensial (persamaan 5.5), karena kedua suku kontrol memiliki persamaan dinamis, maka versi diskretnya hanya dapat diperoleh dengan menggunakan pendekatan numerik.

Salah satu pendekatan yang dapat digunakan pada kedua persamaan tersebut adalah metode numerik rectangular mundur (*backward rectangular*) berikut:

Pendekatan untuk persaman integrator:

$$CO_I(t) = \frac{K_P}{T_I} \int_0^t e(\tau) d\tau \approx CO_I(k) = \frac{K_P}{T_I} \sum_{i=0}^k e(i) T_c \quad (5.7)$$

atau dapat ditulis:

$$CO_I(k) = \frac{K_P T_c}{T_I} \sum_{i=0}^k e(i) = \frac{K_P T_c}{T_I} [e(0) + e(1) + e(2) + \dots + e(k-1) + e(k)] \quad (5.8)$$

Berdasarkan persamaan 5.8, nampak bahwa output Integrator pada cuplikan ke  $-k$  selain tergantung pada error saat itu, juga tergantung dari error-error terdahulu (waktu cuplikan sebelumnya). Jika persamaan diatas diimplementasikan langsung dalam program, maka sistem kontrol membutuhkan memori yang tidak terbatas untuk menyimpan nilai error terdahulu tersebut. Sehingga dengan demikian agar output kontrol integral dapat dihitung secara efisien, maka implementasinya dalam program harus direalisasikan secara rekursif.

Dalam hal ini, komputasi rekursif membutuhkan pengetahuan nilai sinyal output suku integral pada pencuplikan sebelumnya (lihat persamaan 5.9 dibawah)

$$CO_I(k-1) = \frac{K_P T_c}{T_I} \sum_{i=0}^{k-1} e(i) = \frac{K_P T_c}{T_I} [e(0) + e(1) + e(2) + \dots + e(k-1)] \quad (5.9)$$

Dengan mensubstitusikan persamaan 5.9 kedalam persamaan 5.8, maka sekarang output integrator dapat dirumuskan seperti nampak pada persamaan 5.10.

$$CO_I(k) = CO_I(k-1) + \frac{K_P T_c}{T_I} e(k) \quad (5.10)$$

Berdasarkan persamaan 5.10 diatas, terlihat bahwa output integrator pada waktu cuplik ke  $-k$ , sekarang dapat dihitung berdasarkan error saat itu ditambah dengan sinyal output integrator waktu cuplik sebelumnya.

Sedangkan untuk kontrol Diferensial, berdasarkan pendekatan metode *backward difference*, output pada waktu cuplik ke- $k$  dapat dihitung berdasarkan selisih error saat itu dengan error cuplikan sebelumnya (lihat persamaan 5.11).

$$CO_D(k) = K_P T_D \frac{e(k) - e(k-1)}{T_c} \quad (5.11)$$

Sehingga dengan mengacu persamaan 5.2, PID ideal versi diskret yang ditanam dalam sistem digital akan memiliki bentuk seperti nampak pada persamaan 5.12.

$$CO(k) = CO_P(k) + CO_I(k) + CO_D(k) \quad (5.12)$$

dalam hal ini:

$$CO_P(k) = K_p e(k)$$

$$CO_I(k) = CO_I(k-1) + \frac{K_p T_c}{T_I} e(k)$$

$$CO_D(k) = K_p T_D \frac{e(k) - e(k-1)}{T_c}$$

Listing 1 berikut memperlihatkan dua buah fungsi dalam bahasa pemrograman C untuk merealisasikan algoritma kontrol PID-persamaan 5.12 diatas dalam sistem microprocessor: (1) `init_pid` digunakan untuk menginisialisasi parameter PID dan (2) `output_pid` digunakan untuk menghitung besar sinyal kontrol PID.

**//listing 1:**

**//fungsi inialisasi parameter PID**

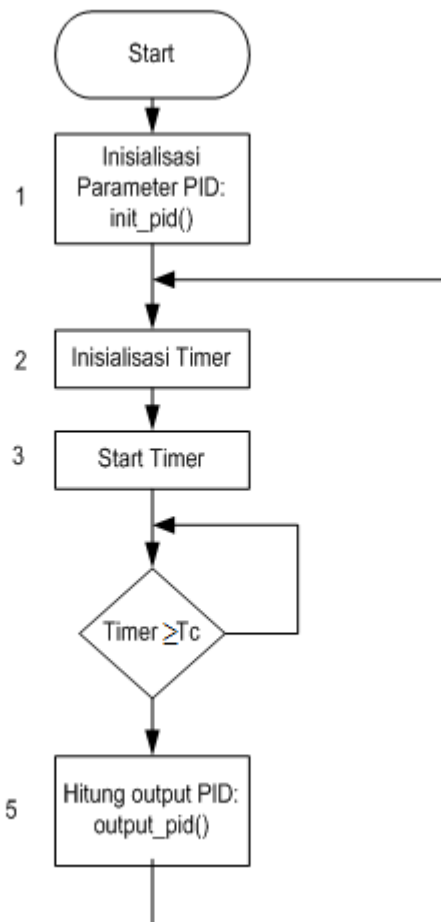
```
void init_pid(float gain_prop, float time_int, float time_def, float time_step)
{
    kp=gain_prop;
    ti=time_int;
    td=time_def;
    tc=time_step;
}
```

**//fungsi kontrol PID tipe ideal-persamaan 5.12**

```
float output_pid(float error)
{
    co_p=kp*error; //hitung output kontrol Proporsional
    co_i=co_i+(kp*tc*error)/ti; //hitung output integral
    co_d=kp*td*(error -prev_error)/tc; //hitung output diferensial
    co=co_p+co_i+co_d; //hitung output PID
    prev_error=error; //geser error
    return co;
}
```

Karena parameter kontrol PID (`kp`, `ti`, `td`, dan `tc`) harus dapat diakses oleh kedua fungsi `init_pid` dan `output_pid`, maka dalam program utama, parameter kontrol tersebut perlu dideklarasikan sebagai variabel global. Gambar 5.3 memperlihatkan *flowchart* pemanggilan kedua fungsi pada program utama.

Seperti nampak pada *flowchart*, dalam program utama, fungsi `output_pid` harus selalu dipanggil setiap waktu cuplik yang telah ditentukan sebelumnya. Sedangkan untuk fungsi `init_pid` pemanggilan dalam program utama dilakukan satu kali yaitu sebelum masuk pada looping pengontrolan (atau dapat juga dipanggil saat operator melakukan pergantian parameter kontrol PID).



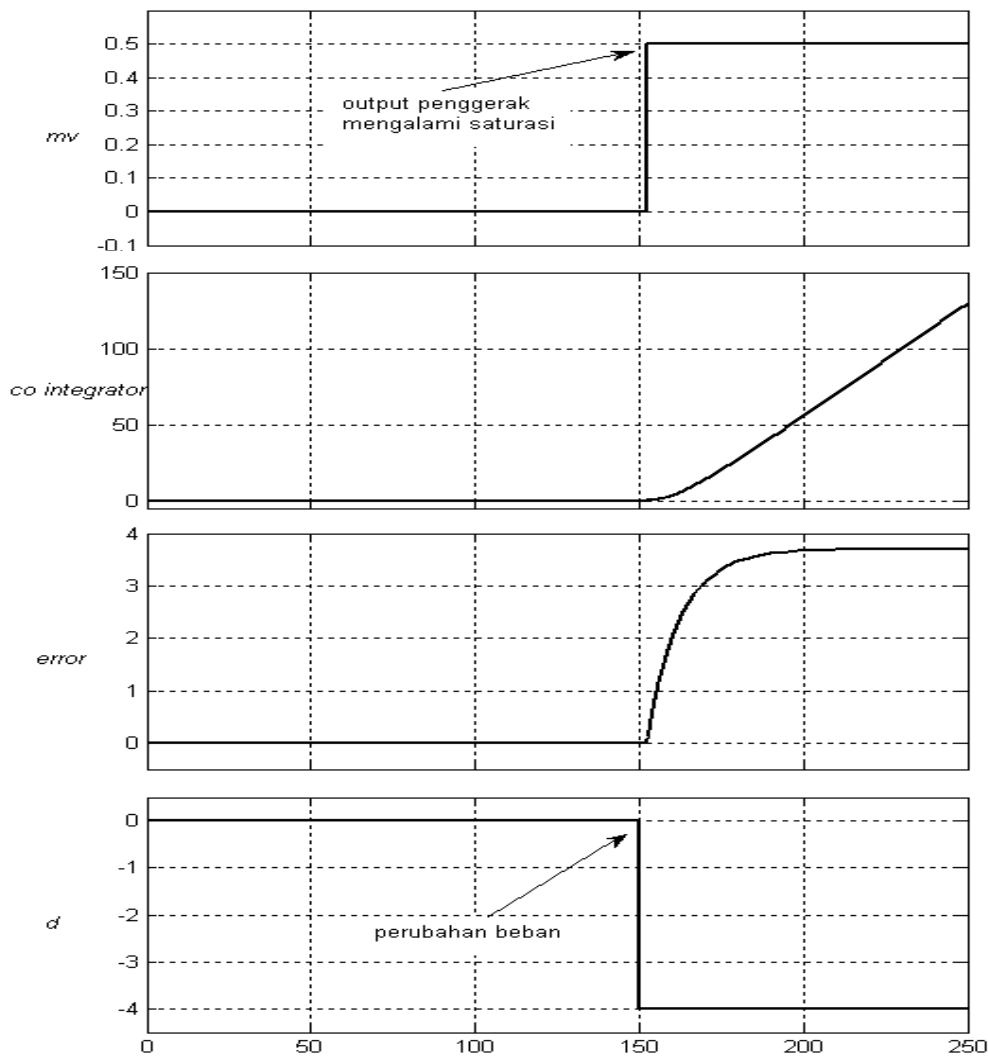
Gambar 5.3. Flowchart kontrol PID digital

### 5.3 Anti Wind Up dan Realisasi Program

Seperti yang telah dijelaskan pada **Bab 2**, fenomena integrator *wind up* dapat terjadi karena adanya batas kemampuan penggerak dalam menanggapi besar sinyal kontrol. Phenomena ini disebabkan karena beban proses yang dikontrol sudah diluar kemampuan penggerak tersebut. Dalam hal ini, walaupun dalam proses kontrol masih terdapat error (selisih antara *SP* dan *PV*), penggerak sudah tidak dapat menggerakkan beban lagi (sudah mengalami saturasi) padahal pada satu sisi berdasarkan sifatnya, output suku integrator pada sistem kontrol PID masih terus-menerus membesar atau terus menerus mengecil-*wind up* (lihat ilustrasinya pada gambar 5.4).

Jika misal pada satu saat beban yang dikontrol kembali menjadi normal, maka output kontroler akan memerlukan waktu yang relative lama untuk menuju nilai nominalnya kembali.

Dalam sistem kontrol, kejadian seperti diatas selain akan menurunkan performansi pengontrolan, sistem dimungkinkan juga menjadi tidak stabil.



Gambar 5.4. Phenomena Windup

Salah satu solusi sederhana untuk menghindari fenomena *WindUp* diatas yaitu dengan cara menghentikan integrasi sinyal kontrol sesaat setelah sistem mengalami saturasi (lihat listing 2 dibawah).

**//listing 2**

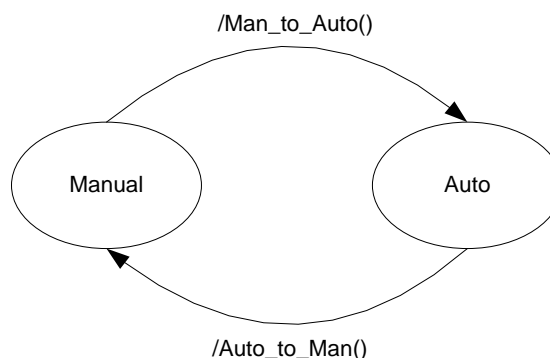
```
// fungsi output PID dengan integrator windup
float output_pid(void)
{
    co_p=kp*error;                               //hitung output kontrol Proporsional
    if(co_i>MAX)
    {
        co_i=MAX;                                //jika terjadi saturasi maksimum
    }
    if(co_i<=MIN)
```

```
{
    co_i=MIN;           //jika terjadi saturasi minimum
}

else
{
    co_i=co_i+(kp*tc*error)/ti; //hitung output integral
}
co_d=kp*td*(e -prev_error)/tc; //hitung output diferensial
co=co_p+co_i+co_d;          //hitung output PID
prev_error=error;           //geser error
return co;
}
```

#### 5.4 Bumpless transfer dan Realisasi Program

Dalam pengontrolan proses, pengubahan mode kontrol dari Manual ke Auto atau sebaliknya adalah sesuatu yang umum dilakukan (lihat diagram state pada gambar 5.5). Satu hal penting yang perlu diperhatikan ketika akan merealisasikan hal tersebut yaituantisipasi lonjakan sinyal kontrol sesaat setelah transfer mode dilakukan. Lonjakan dapat terjadi karena dimungkinkan adanya perbedaan nilai awal output kontrol yang dimiliki pada masing-masing mode.



**Gambar 5.5.** Pengubahan Mode Kontrol

Untuk menghindari lonjakan sinyal kontrol, sesaat setelah peralihan mode dari Manual ke Auto atau sebaliknya, besar sinyal output kontrol awal pada mode kontrol yang dituju perlu diinisialisasi dengan besar sinyal kontrol mode yang ditinggalkan. Listing 3 berikut memperlihatkan realisasi fungsi sederhana untuk meminimalkan lonjakan sinyal kontrol saat peralihan mode.

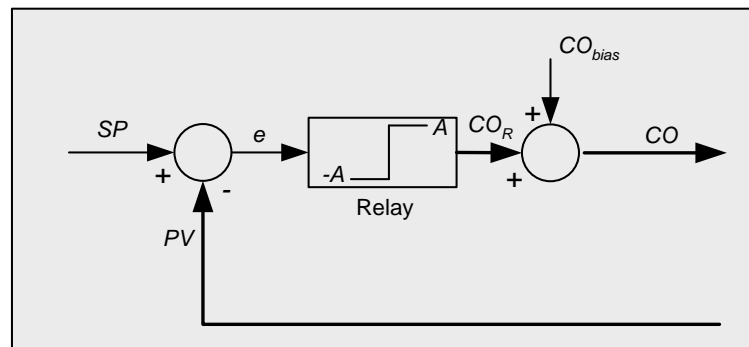
```
//listing 3
//realisasi bumpless transfer
void manual_to_auto(void)
{
    sp=pv; //inisialisasi sp dengan nilai proses variabel sekarang
    co_i=co_manual //inisialisasi kontrol integral dengan kontrol manual
}
```

```
void auto_to_manual(void)
{
    sp=pv; //inisialisasi sp dengan nilai proses variabel sekarang
    co_manual =co_i; //inisialisasi output kontrol manual dengan kontrol integral
}
```

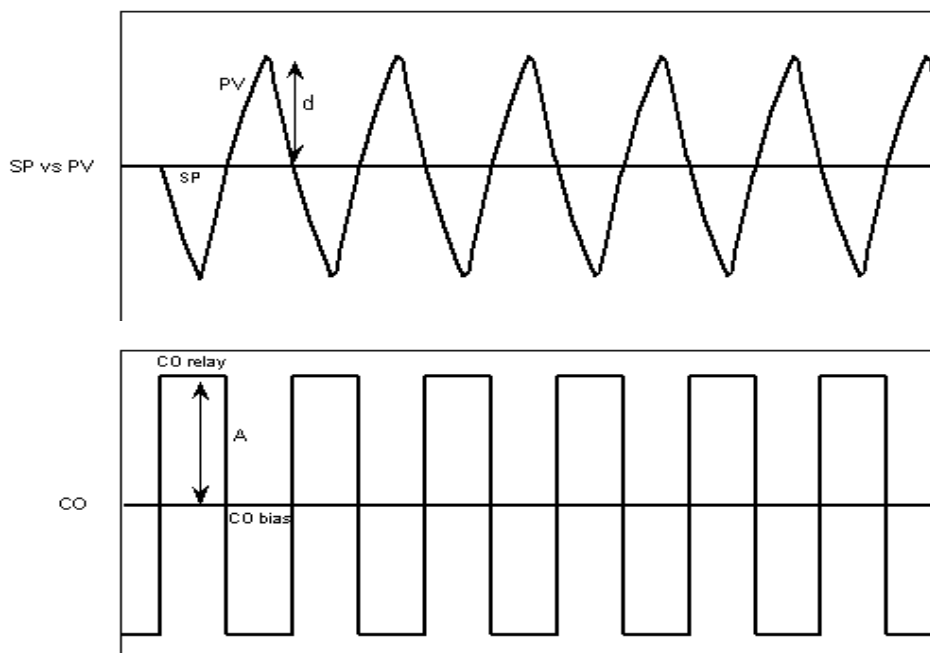
### 5.5 Sistem Kontrol Umpan Balik Relay dan Realisasi Program

Jika output variabel proses yang dikontrol tidak dituntut untuk sama persis dengan nilai Setpoint, maka sistem kontrol umpan balik relay pada dasarnya sudah cukup memadai digunakan sebagai basis pengontrolan variabel proses.

Gambar 5.6 dan 5.7 berikut berturut-turut memperlihatkan diagram blok sistem kontrol umpan balik relay dan hubungan antara sinyal kontrol vs output proses tipikal yang dihasilkan oleh sistem kontrol tersebut.



**Gambar 5.6.** Diagram blok sistem kontrol umpan balik relay

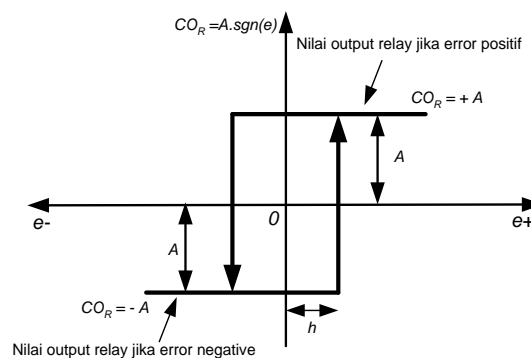


**Gambar 5.7.** Hubungan tipikal output kontrol relay dengan variabel proses

Dalam program, sistem kontrol relay tersebut dapat direalisasikan dalam bentuk fungsi seperti terlihat pada listing 4 berikut:

**//listing 4**

```
// fungsi kontrol umpan balik relay
float output_relay(float error)
{
    float output_co_relay;
    output_co_relay = magnitude_relay*sgn(error)+co_bias;
    return output_co_relay
}
//atau
float output_relay(float error)
{
    float output_co_relay;
    if(error>0)
        output_co_relay = magnitude_relay+co_bias;
    else
        output_co_relay = -magnitude_relay+co_bias;
    return output_co_relay
}
```



**Gambar 5.8.** Hubungan error denga output kontrol Relay plus hysteresis

Karena output proses sesungguhnya akan menunjukkan derau disekitar nilai nominalnya, maka untuk menghindari pengaruh derau pengukuran terhadap peralihan sinyal kontrol, pada output relay tersebut perlu ditambahkan histerisis dengan lebar disesuaikan terhadap level derau yang terukur (lihat listing 5 dibawah)

**//listing 5**

```
// fungsi kontrol umpan balik relay plus hysteresis
float output_relay(void)
{
    float output_co_relay;
    if(error>h) //h : lebar histerisis
        output_co_relay = magnitude_relay+co_bias;
    if(error<-h)
        output_co_relay = -magnitude_relay+co_bias;
}
```



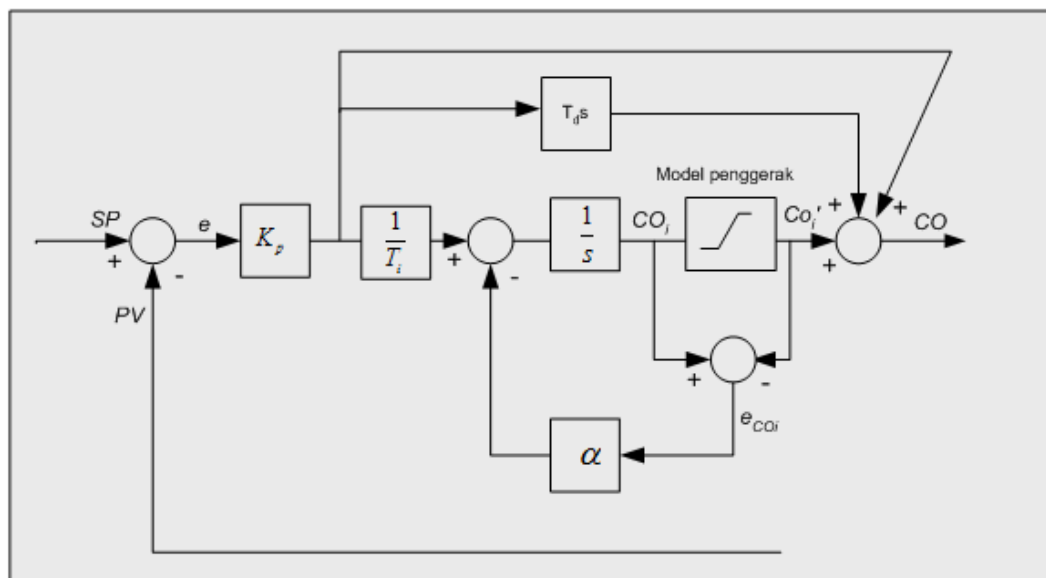
```
return output_co_relay
}
```

## 5.5. Contoh Soal dan Penyelesaian

### Contoh 5.1

#### Realisasi alternative Anti WindUp

Gambar 5.9 berikut memperlihatkan blok diagram kontrol PID dengan antiwindup alternative seperti yang telah dijelaskan pada **Bab 2**.



**Gambar 5.9.** Kontrol PID dengan Antiwindup alternative

#### Pertanyaan

Dengan mengacu gambar tersebut, realisasikan algoritma kontrol PID versi diskretnya dalam bahasa pemrograman C.

#### Penyelesaian

Dengan mengacu pada gambar 5.9 diatas, output kontrol PID dapat dirumuskan:

$$CO(s) = K_p e(s) + CO_i'(s) + K_p T_d s e(s) \quad (5.13)$$

Dalam hal ini  $CO_i'(s)$  adalah output aktual dari blok kontrol integrator:

$$CO_i'(s) = \frac{1}{s} y(s) \quad (5.14)$$

dengan

$$y(s) = \left[ \frac{K_P}{T_i} e(s) - \alpha (CO_i(s) - CO_i'(s)) \right] \quad (5.15)$$

Berdasarkan persamaan 5.14 tersebut, nampak bahwa jika output integrator hasil perhitungan ( $CO_i$ ) belum mengalami saturasi ( $CO_i(s) = CO_i'(s)$ ) maka persamaan output integrator aktual akan direduksi menjadi output integrator normal:

$$CO_i'(s) = \frac{K_P}{T_i s} [e(s)] \quad (5.16)$$

Sedangkan jika terjadi saturasi ( $CO_i \geq \text{MAX}$  atau  $CO_i \leq \text{MIN}$ ) maka output integrator aktual akan direset oleh suku  $\alpha (CO_i(s) - CO_i'(s))$ .

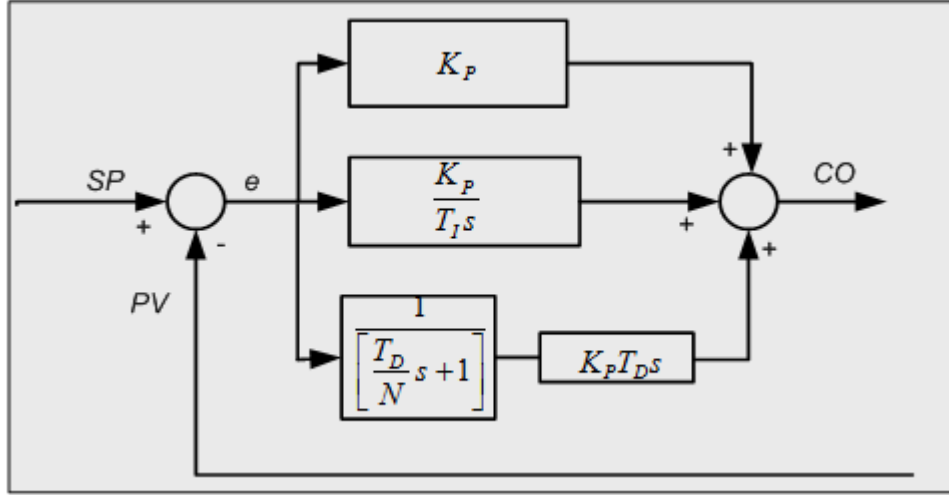
Listing 6 berikut memperlihatkan realisasi program untuk blok diagram gambar 5.9 diatas.

```
//listing 6
float output_pid(float error)
{
    co_p=kp*error;           //hitung output kontrol Proporsional
    if(co_i>=MAX)             //jika terjadi saturasi MAX
    {
        y=e-alpha*(co_i-co_i_aksen);
        co_i_aksen=co_i_aksen+(kp*tc*y)/ti;
    }
    if(co_i<=MIN)            //terjadi saturasi MIN
    {
        y=e-alpha*(co_i-co_i_aksen);
        co_i_aksen=co_i_aksen+(kp*tc*y)/ti;
    }
    else
    {
        co_i=co_i+(kp*tc*error)/ti;
        co_i_aksen=co_i;
    }
    co_d=kp*d*(e-prev_error)/tc; //hitung output diferensial
    co=co_p+co_i_aksen+co_d;      //hitung output PID
    prev_error=error;            //geser error
    return co;
}
```

### Contoh 5.2

#### PIDF Versi diskret

Tinjau diagram blok kontrol PID dengan filter orde 1 (PIDF) pada gambar 5.10



**Gambar 5.10.** Diagram blok kontrol PID (ideal) dengan filter orde 1

### Pertanyaan

Dengan mengacu gambar 5.10 diatas, cari veri diskret untuk PIDF tersebut

### Penyelesaian

Seperti nampak pada gambar 5.10, relasi output kontrol terhadap sinyal error dalam kawasan laplace:

$$CO(s) = K_p \left[ 1 + \frac{1}{T_i s} + \frac{T_d s}{\frac{T_d}{N} s + 1} \right] E(s) \quad (5.17)$$

Dalam kawasan waktu nyata, persamaan 2.17 dapat ditulis:

$$CO(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de_f(t)}{dt} \right] \quad (5.18)$$

dengan

$$\frac{T_d}{N} \frac{de_f(t)}{dt} + e_f(t) = e(t) \quad (5.19)$$

Analog dengan diskritisasi PID ideal, versi diskret PID persamaan 5.18 diatas dapat diperoleh dengan terlebih dulu mencari versi deskrit masing-masing suku penyusun PIDF tersebut:

$$CO(k) = CO_p(k) + CO_i(k) + CO_d(k) \quad (5.12)$$

dengan:

$$CO_p(k) = K_p e(k)$$

$$CO_i(k) = CO_i(k-1) + \frac{K_p T_c}{T_i} e(k)$$

$$CO_d(k) = K_p T_D \frac{e_f(k) - e_f(k-1)}{T_c}$$

dalam hal ini  $e_f(k)$  dapat dicari dari deskritisasi persamaan 5.19:

$$\frac{T_D}{N} \left[ \frac{e_f(k) - e_f(k-1)}{T_c} \right] + e_f(k) = e(k)$$

atau :

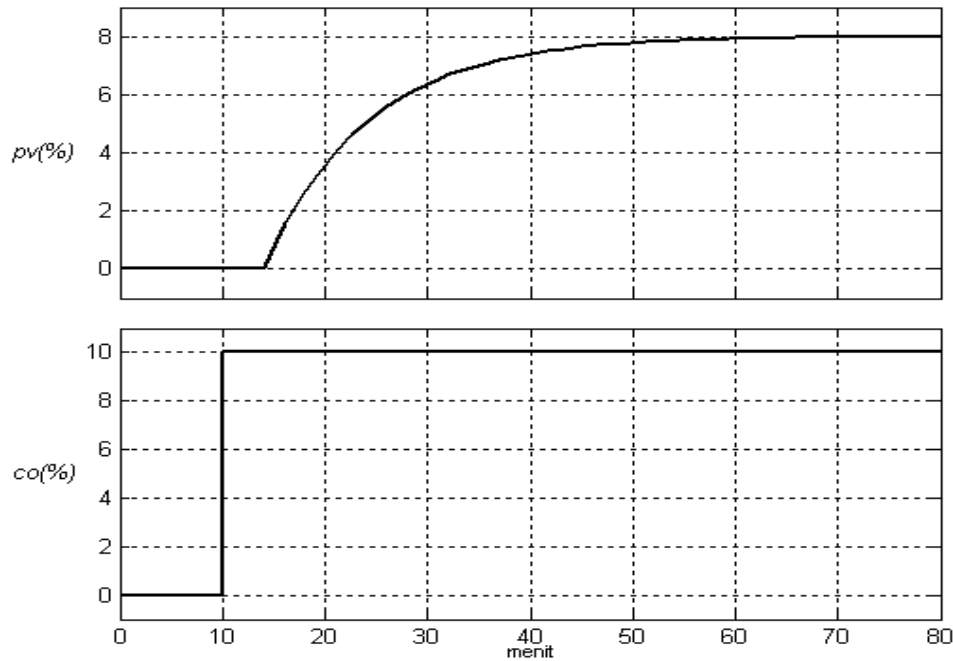
$$e_f(k) = \frac{T_D}{T_c N + T_D} e_f(k-1) + \frac{T_c N}{T_c N + T_D} e(k)$$

## 5.6 Ringkasan

- Sistem kontrol PID digital adalah sistem kontrol PID berbasis sistem microprosesor. Dalam bentuk matematisnya, PID digital akan memiliki bentuk persamaan matematika diskret.
- Waktu cuplik maksimal yang digunakan pada kontrol PID digital sangat tergantung dari dinamika proses yang dikontrol: semakin cepat dinamika sebuah proses, waktu cuplik yang digunakan harus semakin cepat juga (waktu cuplik yang digunakan minimal sebesar 0.1 konstanta waktu proses)
- Cara termudah untuk mendapatkan model matematis PID digital adalah dengan deskritisasi versi PID analog-nya. keakuratan komputasi PID digital sangat tergantung dari waktu cuplik yang digunakan: Semakin sempit waktu cuplik, keakuratannya akan semakin baik.
- Dengan menggunakan sistem digital, Antiwindup dalam sistem kontrol PID dapat direalisasikan secara sederhana yaitu dengan cara menghentikan integrasi sinyal kontrol ketika hasil output kontrol PID telah lebih besar dari batas saturasi penggeraknya.

## 5.7 Soal Latihan

**5.1.** Gambar 5.11 berikut adalah hubungan input-output proses berdasarkan eksperimen Bump test. Berdasarkan grafik tersebut, berpakah waktu cuplik kontrol digital yang tepat untuk pengontrolan proses tersebut



**Gambar 5.11.** Diagram blok kontrol PID (ideal) dengan filter orde 1

**5.2.** Dengan menggunakan metode *backward difference*, carilah versi diskret dari PID berikut

a. PID tipe B:

$$CO(s) = K_p \left[ 1 + \frac{1}{T_I s} \right] e(s) - K_p T_D s PV(s)$$

b. PID tipe C:

$$CO(s) = K_p \left[ -PV(s) + \frac{e(s)}{T_I s} - T_D s PV(s) \right]$$

c. PID general

$$CO(s) = K_p \left[ (\alpha SP(s) - PV(s)) + \frac{E(s)}{T_I s} + \frac{T_D s}{\frac{T_D}{N} s + 1} [\gamma SP(s) - PV(s)] \right]$$

**5.3.** Realisasikan hasil jawab 5.2 diatas dalam bentuk fungsi yang ditulis dalam bahasa pemrograman C.

**5.4.** Realisasikan algoritma kontrol PIDF hasil penyelesaian contoh soal 5.2 dalam bahasa pemrograman C.

## Daftar Pustaka

1. Zoran Vukic, PhD, “*Lectures On PID Controllers*”, University of Zagreb, April 2002
2. Coleman Brosilow, Babu Joseph, “ *Techniques of Model-Based Control*”, Prentice Hall 2002
3. Karl Johan Astrom, Bjorn Wittenmark, “ *Adaptive Control*”, Addison-Wesley Publishing Company, 1995
4. Karl Johan Astrom, Bjorn Wittenmark, “*Computer- Controlled Systems Theory and Design*”, Prentice Hall 1997
5. A.O. Dwyner, “*PI and PID Tuning Rules for Time Delay Processes: a Summary*”, Technical Report AOD-00-01, Edition I, May 2000
6. Carl Knospe, “*PID Control*”, IEEE Control Systems Magazine, Vol. 26.Number 3, February 2006
7. Yun Li, Kiam Heong Ang, C.Y Chong, “*PID Control System Analysis and Design*”, IEEE Control Systems Magazine, Vol. 26.Number 3, February 2006
8. Yun Li, Kiam Heong Ang, C.Y Chong, “*Patent, Software and Hardware for PID Control*”, IEEE Control Systems Magazine, Vol. 26.Number 3, February 2006
9. Dr. Andy Doonan, Prof. Christ Cox, “*Use of Embedded Applications in Automatic Loop Tuning*”, Proc. Of the NACCQ, 2000.
10. Katsuhiko Ogata, “ *Modern Control Engineering*”, Prentice Hall, 1970
11. Donapati Srinivas, “*Autotuning of PID Controllers*”, Master Thesis of Indian Institute of Technology Bombay, July 2006
12. A. Leva, C Cox, “*Hands On PID Autotuning: A guide to Better Utilasion*”, IFAC Professional Brief, 2000
13. Robert Rice, Douglas Cooper, “*Rule Based Design Methodology for the Control Of Non Self Regulating Processes*”, proc. Of ISA, October 2004.
14. \_\_\_\_\_, “*PID Loop Operation*”, Direct Logic Controller, 2005
15. Veronesi Massimiliano, “*Performance Improvement of Smith Predictor through Automatic Computation of Dead Time*”, Technical Report Yokogawa Italia, Industrial Automation dept. March 3, 2003
16. Weidong Zhang, Xing He, Xiaoming Xu, “*Comparison of Several well-Known Controller used in Process Control*”, Proc. Of ISA transactions 2003
17. Weidong Zhang, Xiaoming Xu, “*Analytical design and analysis of Mismatched Smith Predictor*”, Proc. Of ISA transactions 2001
18. Tushar Chowhan, “*Advance PID Control*” White Paper, ShoftDEL Systems Inc. July 2005

- 
19. -, "AVR211 Discreet PID Control", Application Note ATMEL Microcontroller, 2006
  20. DR. D.J Willis, "Multivariable Control: An Introduction", Lecture Note, November 1999
  21. Daniel R. Lewin, "Introduction To MIMO Control", Lecture Note, Process Control Lab,-
  22. \_\_\_\_\_, "Instruction Manual: Autotune Fuzzy/PID Process Temperature Controller", Temco Manual 2006.
  23. Bob Rice, "Design and Tuning of PID Controller for Integrating (Non Self Regulating) Process", Proc ISA 2002
  24. \_\_\_\_\_, "Control strategy: Ratio Control", Application Note, Techmation, 1999
  25. Peyman Gohari, "Lecture 9: Digital Control Algorithms and Their Implementation", ELEC 6061, 2006
  26. \_\_\_\_\_, "Application Guide PID function", Manual Book Toshiba, 1999
  27. H. Rasmussen, "Automatic Tuning of PID Regulators", Aalborg University, September 2002
  28. \_\_\_\_\_, "Feedforward Control", Fisher-Rosemount System Inc, 2006

**Alamat url:**

<http://bestune.50megs.com>  
<http://www.controlguru.com>  
<http://techteach.no/kybsim/>

## Lampiran A. PENGANTAR TRANSFORMASI LAPLACE

### A.1. Pendahuluan

Transformasi Laplace adalah sebuah metode operasional yang sering digunakan untuk menggambarkan hubungan input-output sistem-sistem dinamis. Dengan menggunakan metode transformasi Laplace, pemecahan persoalan-persoalan dinamika sistem dapat dilakukan relative lebih mudah.

Dalam teori kontrol konvensional, Transformasi laplace ini memegang peranan penting selain sebagai *tool* untuk analisis sistem-sistem dinamis (misal untuk menentukan kestabilan sistem atau plant), juga secara praktis dapat digunakan sebagai *tool* dalam perancangan sistem kontrol umpan balik.

### A.2. Teori Umum Transformasi Laplace

Persamaan A.1 dibawah ini memperlihatkan definisi umum Transformasi Laplace untuk sebuah fungsi dalam kawasan waktu nyata :  $f(t)$ ,  $t \geq 0$ .

$$\mathcal{L}[F(t)] = F(s) = \int_0^{\infty} f(t)e^{-st} dt \quad (\text{A.1})$$

Relasi diatas pada dasarnya memetakan fungsi kawasan waktu kedalam bentuk fungsi kawasan  $s$  (variabel laplace). Dalam hal ini  $F(s)$  adalah transformasi laplace dari  $f(t)$ , dan sebaliknya  $f(t)$  adalah transformasi laplace balik (invers) dari  $F(s)$  atau dapat ditulis:

$$f(t) = \mathcal{L}^{-1}[F(s)] \quad (\text{A.2})$$

Hampir dalam setiap persoalan praktis, kita pada dasarnya tidak perlu lagi memecahkan Persamaan A.1 untuk mendapatkan transformasi laplace sebuah fungsi kawasan waktu yang ingin dicari solusinya. Dalam hal ini, solusi dapat dicari dengan merujuk tabel transformasi laplace yang banyak tersedia pada buku-buku yang membahas persoalan kontrol.

Tabel A.1 dan tabel A.2 dibawah berturut-turut memperlihatkan transformasi laplace untuk beberapa fungsi kawasan waktu yang sering dijumpai pada kasus nyata serta beberapa teorema penting yang terkait dengan transformasi laplace.

**Tabel A.1.** Transformasi Laplace

No	$f(t)$	$F(s)$	Keterangan
1	$\delta(t)$	1	Fungsi Impulsa
2	$I(t)$	$\frac{1}{s}$	Fungsi Step (Fungsi tangga)
3	$t$	$\frac{1}{s^2}$	Fungsi Ramp



4	$e^{-at}$	$\frac{1}{s+a}$	Fungsi Eksponensial
5	$\sin(\omega t)$	$\frac{\omega}{s^2 + \omega^2}$	Fungsi Sinusoidal
6	$\cos(\omega t)$	$\frac{s}{s^2 + \omega^2}$	Fungsi Cosinus

**Tabel A.2.** Beberapa sifat dan teorema penting Transformasi Laplace

No	Sifat/teorema	Keterangan
1	$\mathcal{L}[af_1(t) + bf_2(t)] = aF_1(s) + bF_2(s)$	Teorema kombinasi linear
2	$\mathcal{L}\left[\frac{d}{dt}f(t)\right] = \int_0^\infty \frac{d}{dt}f(t)e^{-st} dt = sF(s) - f(0)$	Teorema Turunan
3	$\mathcal{L}\left[\int_0^t f(\tau) d\tau\right] = \frac{F(s)}{s}$	Teorema Integral
4	$f(0) = \lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} sF(s) \quad t > 0$	Teorema harga awal
5	$f(\infty) = \lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$	Teorema harga akhir
6	$\mathcal{L}[f(t-L)] = e^{-sL} F(s),$	Teorema delay

### Contoh-contoh Soal

#### • Contoh soal A.1.

Tentukan transformasi Laplace untuk fungsi:

$$f(t) = 3e^{-t} - 3^{-2t}$$

### Penyelesaian

Berdasarkan Table A.1, Transformasi Laplace dari masing-masing suku persamaan:

$$\mathcal{L}[e^{-t}] = \frac{1}{s+1}$$

dan

$$\mathcal{L}[e^{-2t}] = \frac{1}{s+2}$$

Maka menurut teorema kombinasi linear pada table A.2:

$$\mathcal{L}[3e^{-t} - e^{-2t}] = 3\mathcal{L}[e^{-t}] - \mathcal{L}[e^{-2t}]$$

$$\begin{aligned} &= \frac{3}{s+1} - \frac{1}{s+2} \\ &= \frac{3(s+2) - (s-1)}{s^2 + 3s + 2} \\ &= \frac{3s + 6 - s - 1}{s^2 + 3s + 2} \\ \mathcal{L}[3e^{-t} - e^{-2t}] &= \frac{2s + 5}{s^2 + 3s + 2} \end{aligned}$$

• **Contoh soal A.2.**

Tentukan transformasi Laplace dari  $\frac{d}{dt}(e^{-t})$

**Penyelesaian**

Berdasarkan teorema turunan pada tabel A.2 :

$$\mathcal{L}\left[\frac{d}{dt}(e^{-t})\right] = sF(s) - f(0)$$

Dalam hal ini :

$$\mathcal{L}[e^{-t}] = \frac{1}{s+1} \quad \text{dan} \quad f(0) = \lim_{t \rightarrow 0} e^{-t} = 1$$

$$\begin{aligned} \text{maka : } \mathcal{L}\left[\frac{d}{dt}(e^{-t})\right] &= sF(s) - f(0) \\ &= s\left(\frac{1}{s+1}\right) - 1 = \frac{s}{s+1} - 1 = \frac{s}{s+1} - \frac{s+1}{s+1} \\ &= \frac{-1}{s+1} \end{aligned}$$

• **Contoh soal A.3.**

Tentukan transformasi Laplace dari :

$$\int_0^t e^{-\tau} d\tau$$

**Penyelesaian**

$$\text{karena } \mathcal{L}[e^{-t}] = \frac{1}{s+1}$$

maka

---

$$\mathcal{L} \left[ \int_0^t f(\tau) d\tau \right] = \frac{F(s)}{s} = \frac{1}{s} \cdot \left( \frac{1}{s+1} \right) = \frac{1}{s(s+1)}$$

- **Contoh soal A.4.**

Tentukan nilai awal dari fungsi  $f(t) = e^{-3t}$

**Penyelesaian**

Dari table A.1 didapat:

$$\mathcal{L} [e^{-3t}] = \frac{1}{s+3}$$

Berdasarkan definisi teorema Harga awal :

$$f(0) = \lim_{t \rightarrow 0} e^{-3t} = \lim_{s \rightarrow \infty} s \left[ \frac{1}{s+3} \right] = \lim_{s \rightarrow \infty} \frac{s/s}{s/s + 3/s} = \frac{1}{1 + \frac{3}{\infty}} = 1$$

- **Contoh soal A.5.**

Tentukan nilai akhir dari fungsi  $f(t) = e^{-3t}$

**Penyelesaian**

Dari table A.1 didapat:

$$\mathcal{L} [e^{-3t}] = \frac{1}{s+3}$$

Berdasarkan definisi teorema Harga akhir:

$$f(\infty) = \lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} s \left[ \frac{1}{(s+3)} \right] = \lim_{s \rightarrow 0} \frac{s}{(s+3)} = \lim_{s \rightarrow 0} \frac{0}{0+3} = 0$$

### A.3. Konsep Fungsi Alih dan Blok Diagram

Fungsi Alih adalah perbandingan transformasi laplace keluaran terhadap transformasi masukan dengan menganggap semua keadaan awal sama dengan nol.

Dalam bidang dinamika dan kontrol, konsep fungsi alih tersebut seringkali digunakan untuk menggambarkan relasi input-output komponen-komponen, *plant* ataupun sistem.

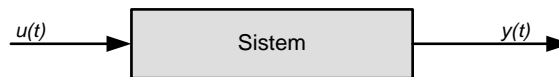
Berbeda dengan representasi persamaan diferensial dalam penggambaran dinamika sistem yang bersifat implisit. Penggambaran dinamika atau hubungan input-output sistem dengan menggunakan konsep fungsi alih ini akan bersifat eksplisit (dengan demikian representasi fungsi alih ini relative mudah dipahami).

- **Contoh soal A.6**

Berikut ini adalah persamaan diferensial yang menggambarkan dinamika sistem orde satu.

$$T \frac{dy(t)}{dt} = -y(t) + K.u(t), \quad y(0) = 0$$

Dalam hal ini  $u(t)$  dan  $y(t)$  berurutan-turut adalah input dan output pada sistem tersebut (lihat gambar A.1 dibawah)



**Gambar A.1.** Hubungan input-output sistem (orde 1) dalam kawasan waktu nyata

Carilah fungsi alih sistem tersebut

### Penyelesaian

Dengan menggunakan teorema diferensial pada Table A.2 dan dengan mentransformasikan masing-masing suku persamaan, maka akan didapat:

$$T[sY(s) - y(0)] = -Y(s) + K.U(s)$$

Dengan mengingat nilai output sistem awal  $y(0)=0$ , dan sedikit langkah penyederhanaan maka hubungan antara input-output sistem tersebut dalam kawasan laplace-nya adalah:

$$Y(s) = \frac{K}{Ts + 1} u(s)$$

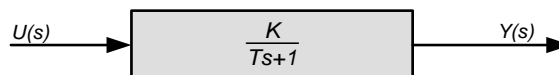
atau dapat ditulis:

$$Y(s) = H(s)u(s)$$

dengan

$$H(s) = \frac{K}{Ts + 1}$$

Dalam hal ini  $H(s)$  adalah fungsi alih (*transfer function*) yang merelasikan antara input-output sistem tersebut. Gambar A.2 dibawah ini memperlihatkan blok diagram untuk persoalan diatas.



**Gambar A.2.** Hubungan input-output sistem (orde 1) dalam kawasan s

### • Contoh soal A.7

Carilah fungsi alih untuk system yang memiliki dinamika berikut

$$T \frac{dy(t)}{dt} = -y(t) + K.u(t - L), \quad y(0) = 0$$

### Penyelesaian

Dengan menggunakan teorema diferensial pada Table A.2 dan dengan mentransformasikan masing-masing suku persamaan, maka akan didapat:

$$T[sY(s) - y(0)] = -Y(s) + Ke^{-Ls}U(s)$$

Dengan mengingat nilai output sistem awal  $y(0)=0$ , dan sedikit langkah penyederhanaan maka hubungan antara input-output sistem tersebut dalam kawasan laplace-nya adalah:

$$Y(s) = \frac{Ke^{-Ls}}{Ts + 1}u(s)$$

atau dapat ditulis:

$$Y(s) = H(s)u(s)$$

dengan

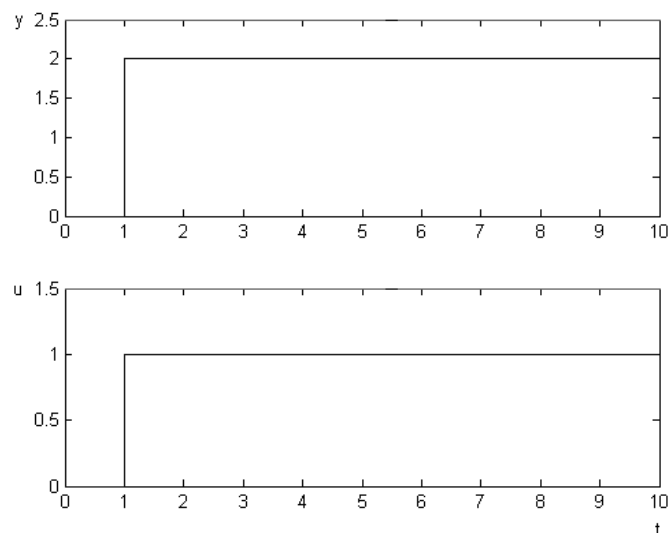
$$H(s) = \frac{Ke^{-Ls}}{Ts + 1}$$

Dalam hal ini  $H(s)$  adalah fungsi alih (*transfer function*) yang merelasikan antara input-output sistem tersebut.

#### A.4. Beberapa Fungsi Alih yang umum dijumpai dalam bidang kontrol proses

- Penguat :  $H(s) = K$

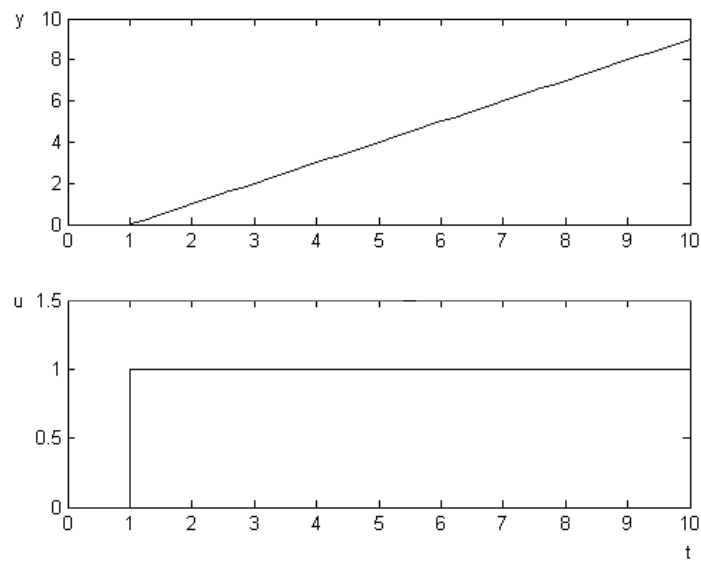
Hubungan input-output sebuah penguat dapat dilihat pada gambar A.3 berikut:



**Gambar A.3.** Hubungan input-output penguat  $H(s) = 2$

- Integrator:  $H(s) = \frac{1}{s}$

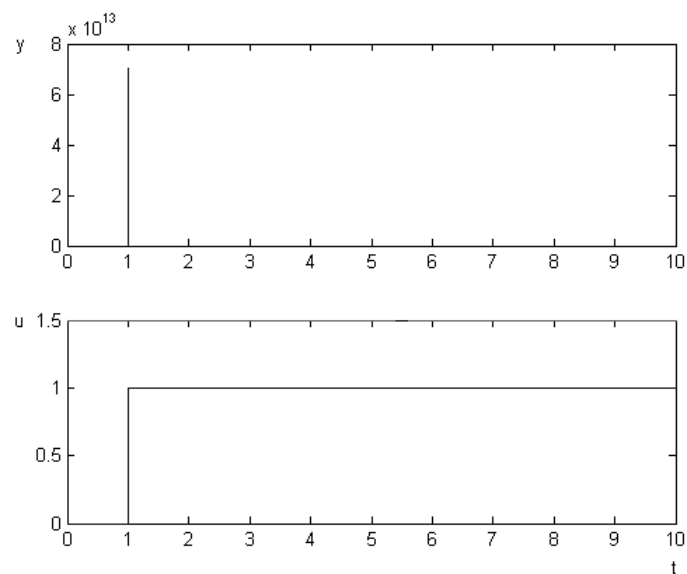
Hubungan input-output sebuah integrator dapat dilihat pada gambar A.4 berikut:



**Gambar A.4.** Hubungan input-output integrator  $H(s) = \frac{1}{s}$

- Diferensiator:  $H(s) = s$

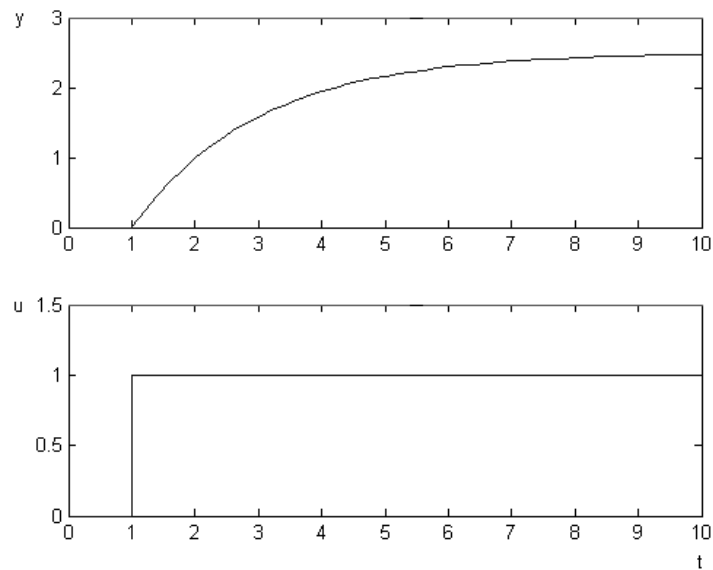
Hubungan input-output sebuah diferensiator dapat dilihat pada gambar A.5 berikut:



**Gambar A.5.** Hubungan input-output diferensiator  $H(s) = s$

- Sistem Orde Satu:  $H(s) = \frac{K}{Ts + 1}$

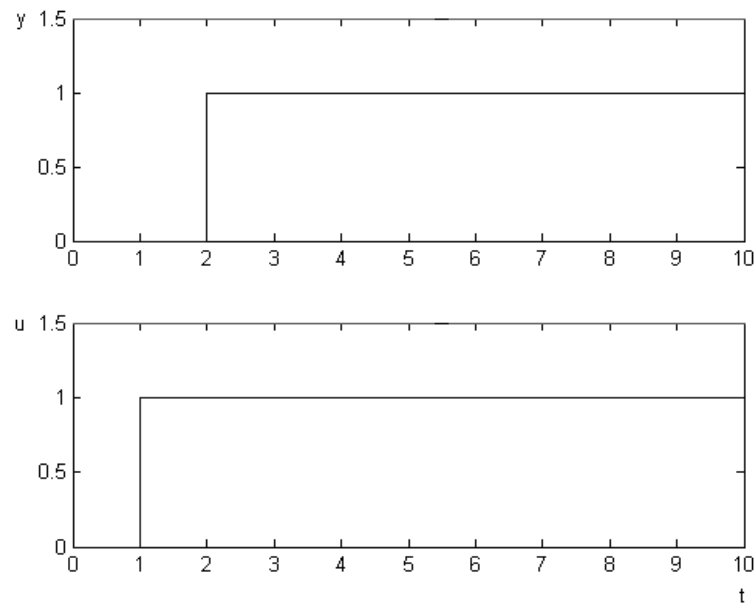
Hubungan input-output system orde satu dapat dilihat pada gambar A.6 berikut:



**Gambar A.6.** Hubungan input-output sistem orde satu  $H(s) = \frac{2.5}{1.5s + 1}$

- Komponen delay:  $H(s) = e^{-Ls}$

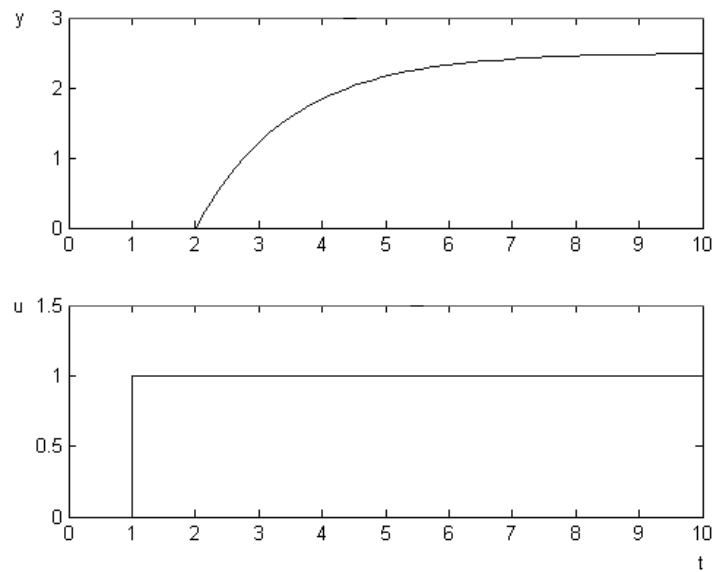
Hubungan input-output komponen delay dapat dilihat pada gambar A.7 berikut:



**Gambar A.7.** Hubungan input-output komponen delay  $H(s) = e^{-s}$

- Sistem Orde Satu plus delay:  $H(s) = \frac{K}{Ts + 1} e^{-Ls}$

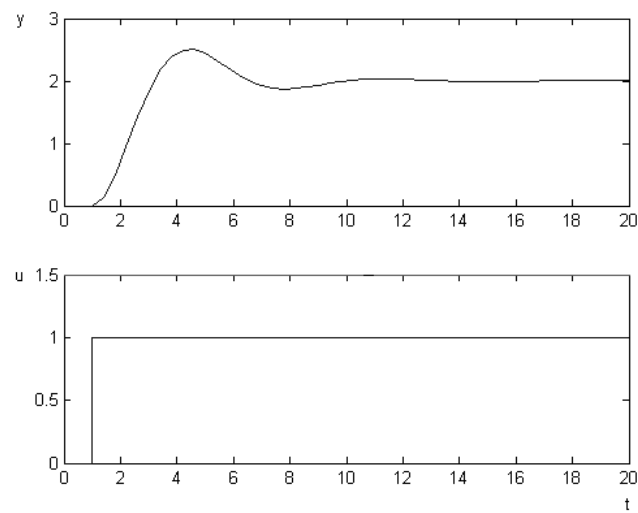
Hubungan input-output Sistem Orde Satu plus delay dapat dilihat pada gambar A.8 berikut:



**Gambar A.7.** Hubungan input-output komponen delay  $H(s) = \frac{2.5}{1.5s+1} e^{-s}$

- Sistem Orde dua:  $H(s) = \frac{K\omega_n}{s^2 + 2\xi\omega_n s + \omega_n^2}$

Hubungan input-output Sistem Orde dua dapat dilihat pada gambar A.9 berikut:



**Gambar A.7.** Hubungan input-output sistem orde dua  $H(s) = \frac{2}{1s^2 + 0.8s + 1}$

ooOoo



---

## LAMPIRAN B

### PENGENALAN MATLAB DAN SIMULINK

#### B.1. Pendahuluan

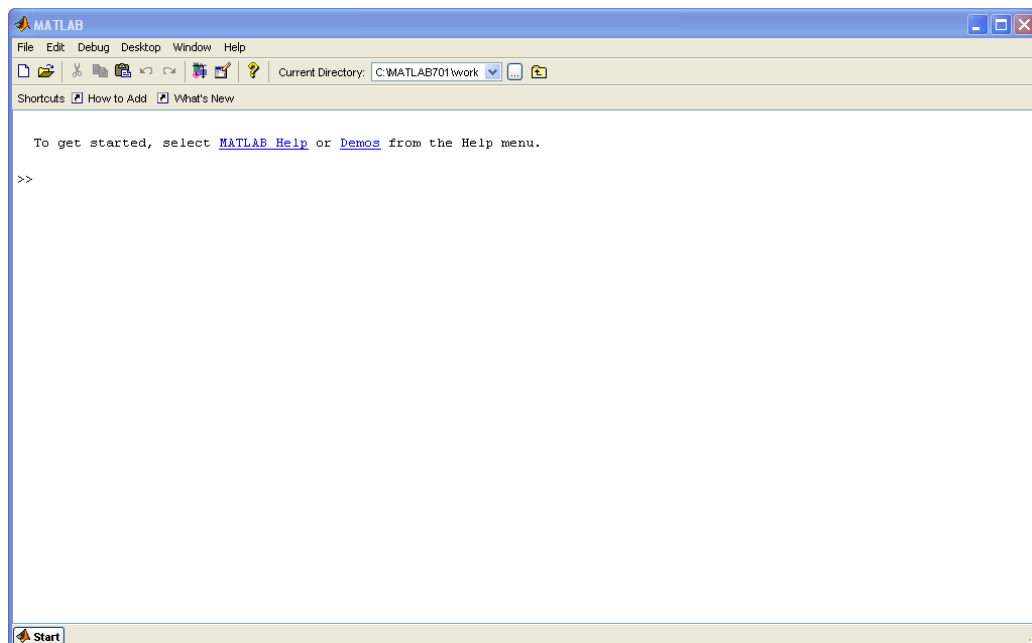
Matlab (Matrix Laboratory) adalah sebuah perangkat lunak keluaran MathWorks Inc. yang pada awalnya ditujukan untuk mempermudah penyelesaian permasalahan-permasalahan komputasi. Dalam Perkembangan selanjutnya, Matlab juga dilengkapi sejumlah fitur dan *toolbox* yang berisi fungsi-fungsi yang sering dijumpai pada berbagai bidang rekaya, statistic, kontrol, ekonomi, pemrosesan sinyal, logika fuzzy dan sebagainya.

Dari sekian banyak fitur yang ditawarkan, salah satunya adalah Simulink. Dengan menggunakan perangkat lunak Simulink ini, perilaku hubungan input-output sistem dinamis dapat disimulasikan secara mudah dengan cara “*click and drag*”

Penggunaan Simulink dalam studi kontrol secara praktis sangat membantu sebagai langkah awal sebelum sistem kontrol *real time*-nya ditanamkan pada sistem microcontroller atau perangkat keras lainnya.

#### B.2. Pengenalan Lingkungan Operasi MATLAB

Gambar B.1 berikut adalah Jendela yang muncul ketika pertama kali perangkat lunak Matlab dijalankan.



Gambar B.1. Jendela Workspace pada Matlab

---

Jendela tersebut dikenal dengan nama *command window* atau *workspace* (ruang kerja). Pada jendela ini, kita dapat melakukan perhitungan-perhitungan matematis serta pemanggilan berbagai macam fungsi yang telah disediakan oleh Matlab.

Sesuai namanya Matrik Laboratory, operasi komputasi dalam Matlab pada dasarnya berbasis manipulasi matrik. Dalam hal ini, perhitungan yang berkaitan dengan matrik, seperti perkalian, penjumlahan, pengurangan, transpose, dan sebagainya sangat mudah dilakukan. misal untuk perkalian dua buah matrik **A** dan **B**, perintah yang perlu diketik dalam jendela workspace cukup : **A\*B** <enter>.

Untuk lebih jelasnya, ketik instruksi –instruksi berikut dalam jendela workspace

```
>> A=[1 3;4 6]    <enter>
A =
     1     3
     4     6
>> B=[4 3;5 3]    <enter>
B =
     4     3
     5     3
>> C=A+B          <enter>
C =
     5     6
     9     9
>> D=A-B          <enter>
D =
    -3     0
    -1     3
>> E=A*D          <enter>
E =
    -6     9
   -18    18
```

### B.3. Pengenalan Toolboxes MATLAB

Sejak awal kelahirannya sampai saat ini, Matlab telah mengalami perkembangan luar biasa terutama terkait dengan fungsi dan operator yang disediakan (built in). untuk memudahkan manajemen penggunaan, fungsi-fungsi tersebut dikelompokkan ke dalam folder atau *toolbox* yang relevan. Misal fungsi-fungsi yang umum dijumpai dalam bidang statistik, seperti max, varian, distribusi poisson dan sebagainya dapat dijumpai pada toolbox stats (statistics).

Untuk mengetahui secara pasti *toolbox-toolbox* apa saja yang tersedia, ketik perintah help pada command window seperti terlihat dibawah:

```
>> help    <enter>
```

HELP topics:

matlab\general	- General purpose commands.
matlab\ops	- Operators and special characters.
matlab\lang	- Programming language constructs.
matlab\elmat	- Elementary matrices and matrix manipulation.
matlab\elfun	- Elementary math functions.
matlab\specfun	- Specialized math functions.
matlab\matfun	- Matrix functions - numerical linear algebra.

---

matlab\datafun	- Data analysis and Fourier transforms.
matlab\polyfun	- Interpolation and polynomials.
matlab\funfun	- Function functions and ODE solvers.
matlab\sparfun	- Sparse matrices.
matlab\graph2d	- Two dimensional graphs.
matlab\graph3d	- Three dimensional graphs.
matlab\specgraph	- Specialized graphs.
matlab\graphics	- Handle Graphics.
matlab\uitools	- Graphical user interface tools.

Sedangkan untuk mengetahui fungsi atau operator yang terdapat pada sebuah toolbox tertentu, maka pada command window ketik help diikuti dengan topik yang dikehendaki.

Contoh:

```
>> help control <enter>
Control System Toolbox
Version 6.1 (R14SP1) 05-Sep-2004

General.
ctrlpref - Set Control System Toolbox preferences.
ltimodels - Detailed help on the various types of LTI models.
ltiprops - Detailed help on available LTI model properties.

Creating linear models.
tf        - Create transfer function models.
zpk       - Create zero/pole/gain models.
ss, dss   - Create state-space models.
frd       - Create a frequency response data models.
filt      - Specify a digital filter.
lti/set   - Set/modify properties of LTI models.

Data extraction.
tfdata    - Extract numerator(s) and denominator(s).
zpkdata   - Extract zero/pole/gain data.
ssdata    - Extract state-space matrices.
dssdata   - Descriptor version of SSDATA.
frdata    - Extract frequency response data.
lti/get    - Access values of LTI model properties.

Conversions.
tf        - Conversion to transfer function.
zpk       - Conversion to zero/pole/gain.
ss        - Conversion to state space.
frd       - Conversion to frequency data.
chgunits  - Change units of FRD model frequency points.
c2d       - Continuous to discrete conversion.
d2c       - Discrete to continuous conversion.
d2d       - Resample discrete-time model.
```

Untuk mengetahui parameter apa saja yang perlu dilewatkan dalam pemanggilan fungsi tertentu, maka ketik help diikuti nama fungsi tersebut

Contoh:

```
>> help sin <enter>
```

---

SIN Sine.  
SIN(X) is the sine of the elements of X.

See also asin, sind.

Overloaded functions or methods (ones with the same name in other directories)  
help sym/sin.m

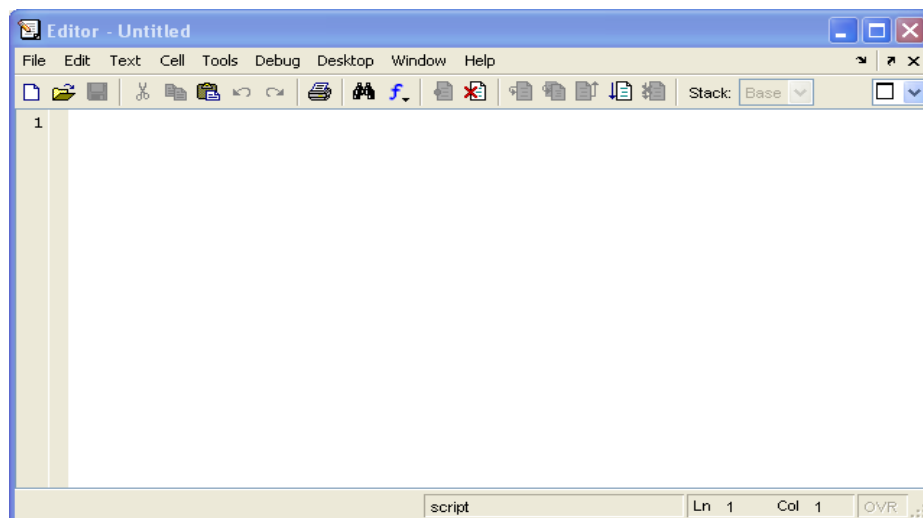
Reference page in Help browser  
doc sin

#### B.4. Matlab sebagai Bahasa Pemrograman

Sebagai bahasa pemrograman, Matlab memungkinkan pembuatan program lengkap dalam sebuah file script berekstensi M file. Program tersebut dapat ditulis pada jendela editor yang telah disediakan oleh matlab.

Berikut adalah langkah-langkah dalam pembuatan program lengkap:

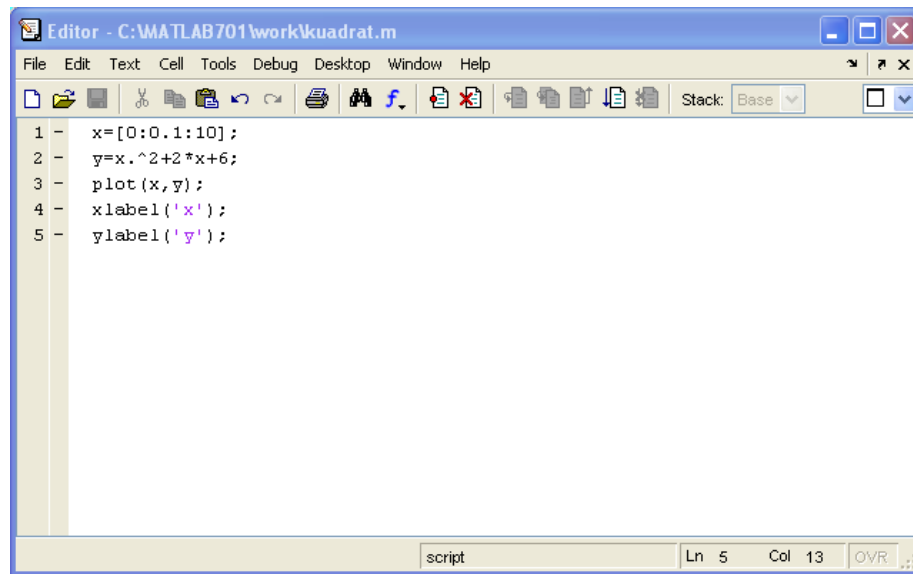
Pada menu utama Command Window, *click* File kemudian pilih sub menu New – M file. Setelah itu, maka akan muncul jendela editor seperti terlihat pada gambar B.2 berikut.



**Gambar B.2.** Jendela Editor matlab

Selanjutnya, ketikkan perintah – perintah program pada jendela tersebut.

Sebagai contoh, ketikkan perintah seperti nampak pada gambar B.3 dibawah, kemudian simpan file tersebut dan beri nama `kuadrat.m`



**Gambar B.3.** Contoh listing untuk plot sebuah grafik

**Keterangan:**

Listing program diatas pada dasarnya berisi perintah untuk mem-plot grafik kuadratis

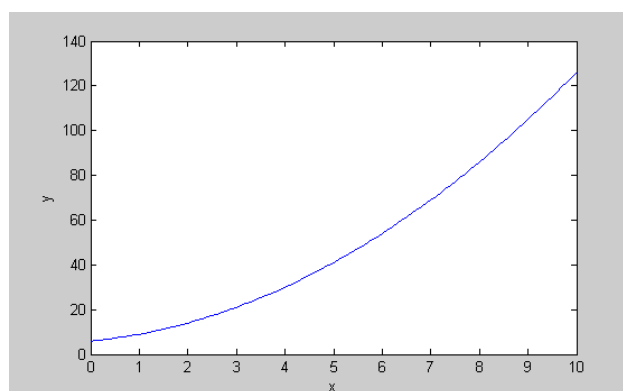
$$y = x^2 + 2x + 6 \quad (\text{lihat baris 2})$$

dimana dalam hal ini jangkauan nilai x adalah 0 sampai 10, dengan beda nilai sebesar 0.1. (lihat baris 1). Perintah menggambar grafik dalam matlab adalah plot yang membutuhkan dua parameter: x dan y (lihat baris 3). Sedangkan dua baris terakhir berfungsi memberikan label pada tiap-tiap sumbunya.

Untuk menjalankan program yang telah dibuat, kembali ke jendela command window, ketik nama program pada prompt dan tekan enter:

>>kuadrat <enter>

maka akan muncul grafik berikut:




**Gambar B.4.** Hasil eksekusi kuadrat.m

### B.5. Pengenalan Lingkungan Operasi Simulink

Simulink adalah sebuah tool yang disediakan Matlab untuk tujuan simulasi, Dengan menggunakan simulink, simulasi cara kerja sistem dinamis dapat dilakukan secara mudah yaitu dengan cara *click and drag* tanpa pengetikan satu baris instruksi pun.

Semua sarana yang dibutuhkan untuk membangun dan mensimulasikan sistem atau proses seperti sumber sinyal, blok fungsi alih, scope dan sebagainya telah tersedia lengkap dalam tool simulink ini

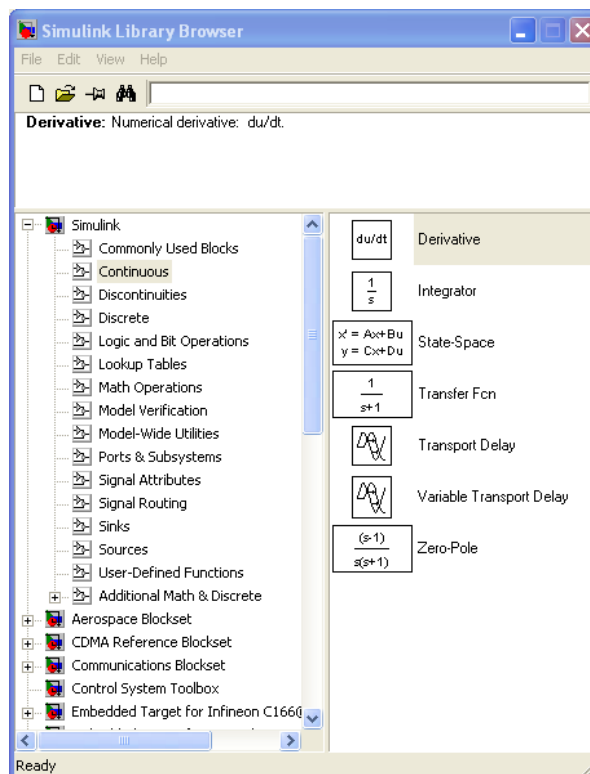
Ada dua cara yang dapat digunakan untuk menjalankan simulink pada Matlab,

1. pada workspace ketik simulink <enter> , atau
2. click icon  yang nampak pada toolbars utama (lihat gambar B.5)



**Gambar B.5.** Toolbar pada Command window Matlab

Jendela yang muncul setelah itu adalah Simulink Library browser seperti nampak pada gambar B.6 berikut

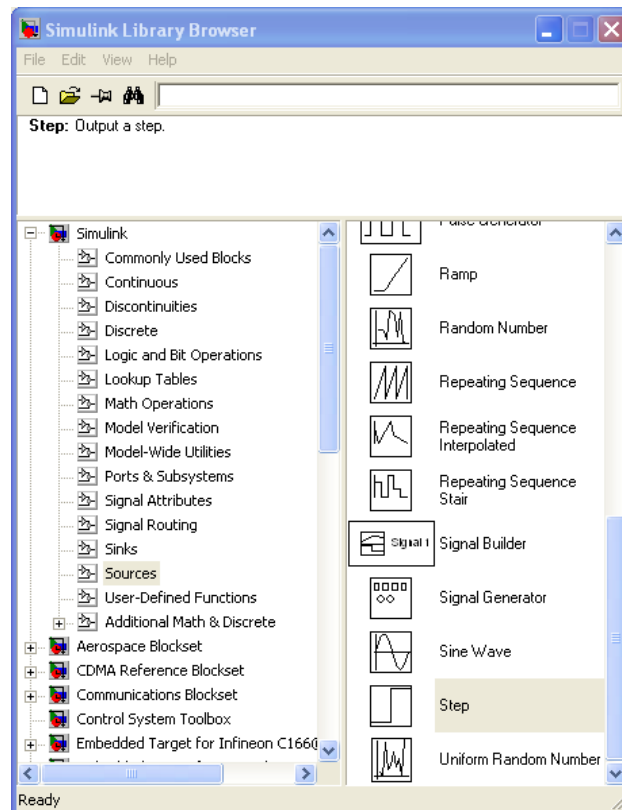


**Gambar B.6.** Jendela Simulink Library browser

Jendela Simulink Library browser ini pada dasarnya adalah jendela utama Simulink. pada jendela ini dapat ditemukan blok-blok fungsional untuk membangun simulasi lengkap.

Agar memudahkan penggunaan, blok-blok fungsional tersebut dikelompokkan menurut katagorinya masing-masing, misal blok yang berkaitan dengan model proses atau plant

kontinyu, seperti blok fungsi alih proses, blok delay transportasi, integrator dan sebagainya dapat ditemukan pada sub-folder continuous (lihat gambar B.6). Untuk blok-blok simulasi input seperti fungsi step, fungsi random, sinus, dan sebagainya dapat dijumpai pada subfolder source (lihat gambar B.7).



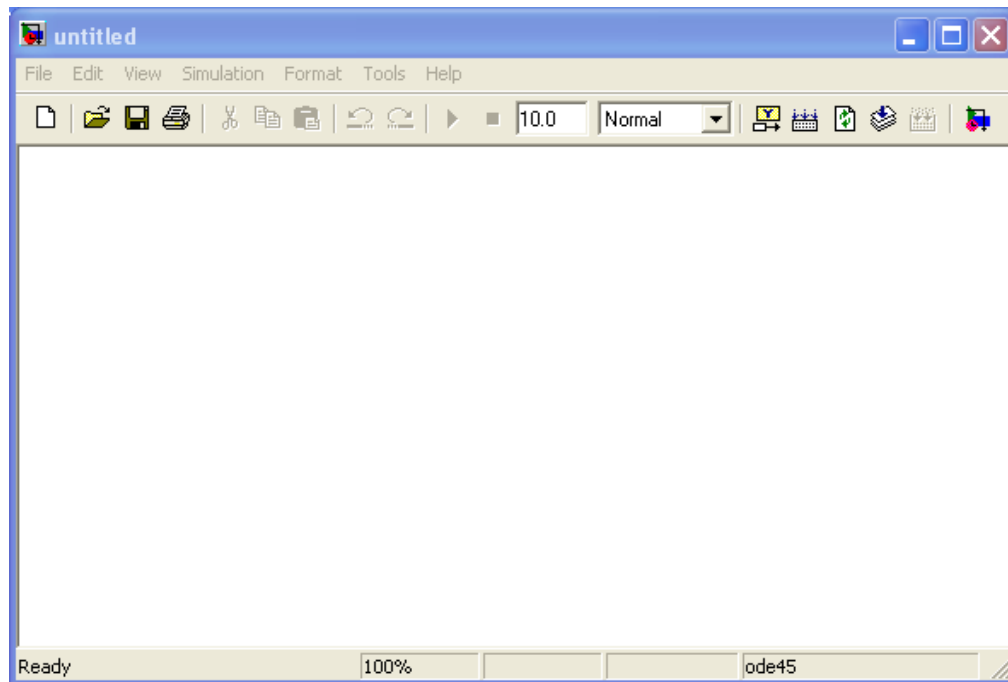
**Gambar B.7.** Subfolder Sources berisi blok-blok input proses

Agar mendapatkan pemahaman yang lengkap, sebaiknya pembaca mencoba langsung mengamati isi subfolder-subfolder lainnya.

Seperti yang telah dijelaskan sebelumnya, untuk membangun sebuah model simulasi, kita sama sekali tidak dituntut untuk menguasai bahasa pemrograman. Hanya dengan metode *click and drag*, simulasi proses dapat dilakukan secara cepat.

Berikut adalah langkah-langkah membangun simulasi proses:

Pada menu utama Simulink Library browser sorot menu file dengan mouse kemudian click sub menu File- model. Maka akan muncul Jendela seperti nampak pada gambar B.8 dibawah



**Gambar B.8.** Jendela tempat simulasi dilakukan

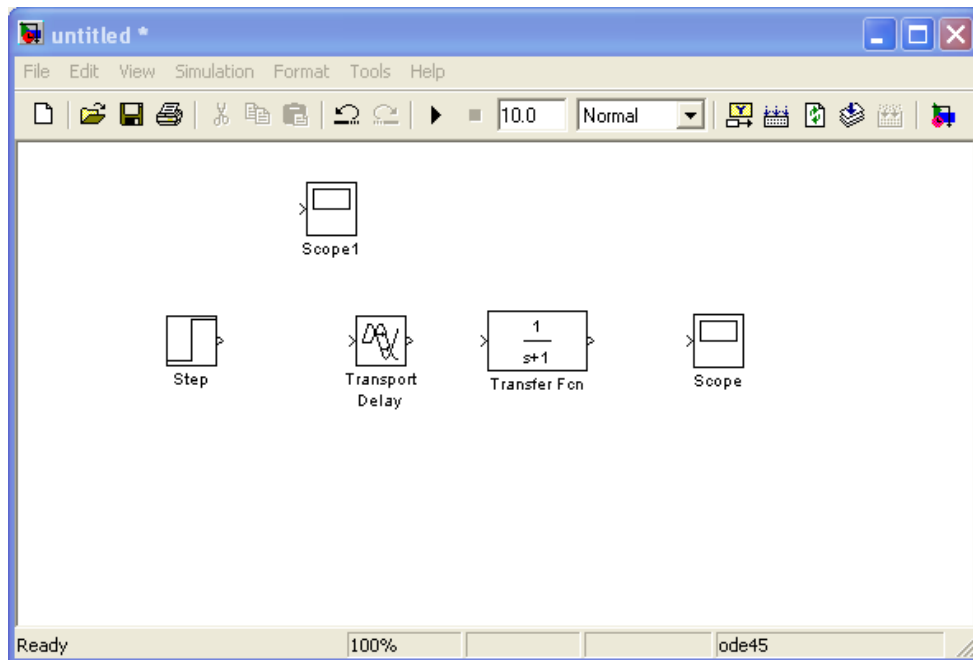
Dengan cara *click and drag*, pindahkan blok-blok model yang tersedia pada Simulink Library Browser kedalam jendela diatas.

Sebagai contoh, berikut adalah langkah untuk mensimulasikan hubungan input-output proses FOPDT dengan fungsi alih:

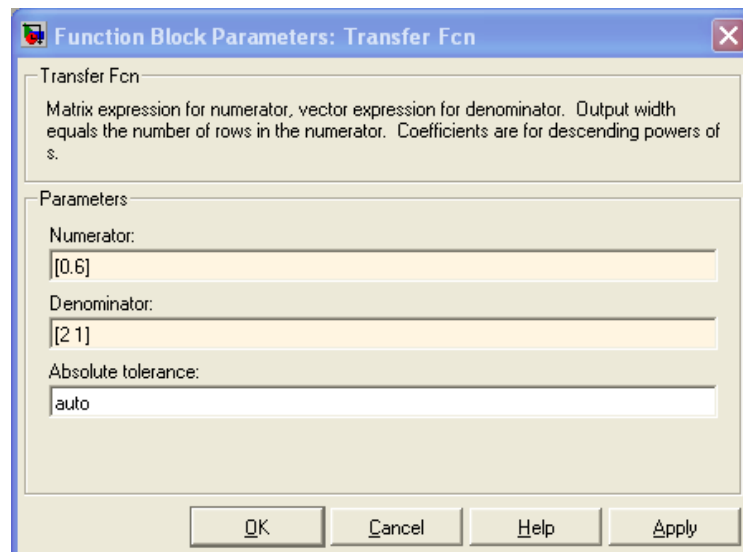
$$H(s) = \frac{0.6}{2s + 1} e^{-s} ; K=0.6, T=2, \text{ dan } L=1 \quad (\text{B.1})$$

1. Pindahkan blok-blok fungsional yang terkait dengan proses tersebut pada jendela Simulasi (lihat gambar B.9)
2. Dengan menggunakan mouse tarik garis untuk menghubungkan relasi fungsional antar blok.
3. Ubah nilai default blok transfer fcn dengan cara men-double click blok tersebut, dan ganti parameter blok dengan nilai seperti tercantum pada persamaan B.1 (lihat gambar B.10). Hal yang sama dilakukan pada blok-blok fungsional lainnya.



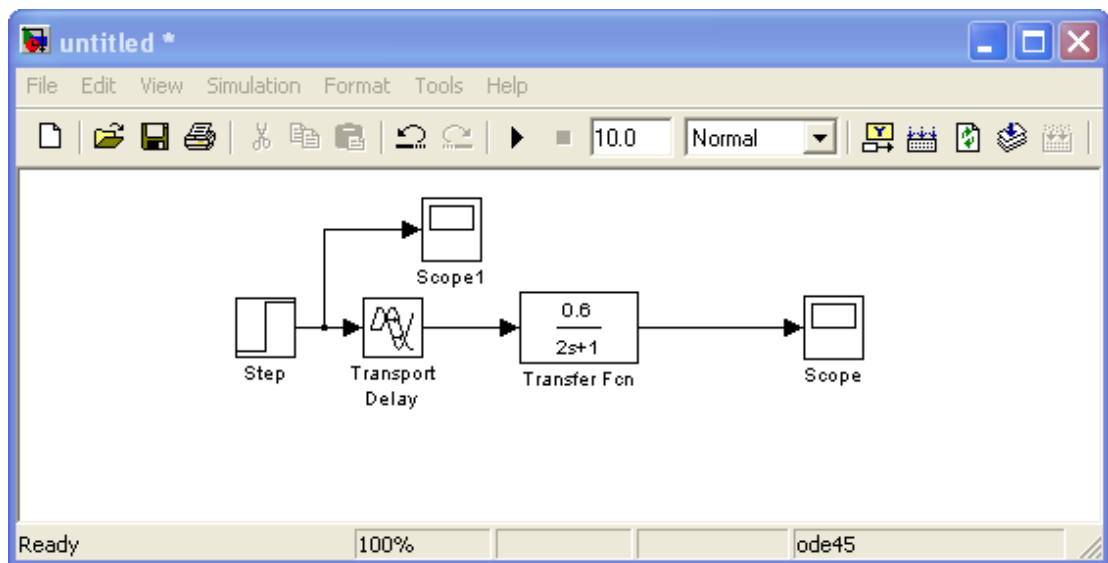


**Gambar B.9.** Blok-blok fungsional hasil click and drag pada jendela Simulasi

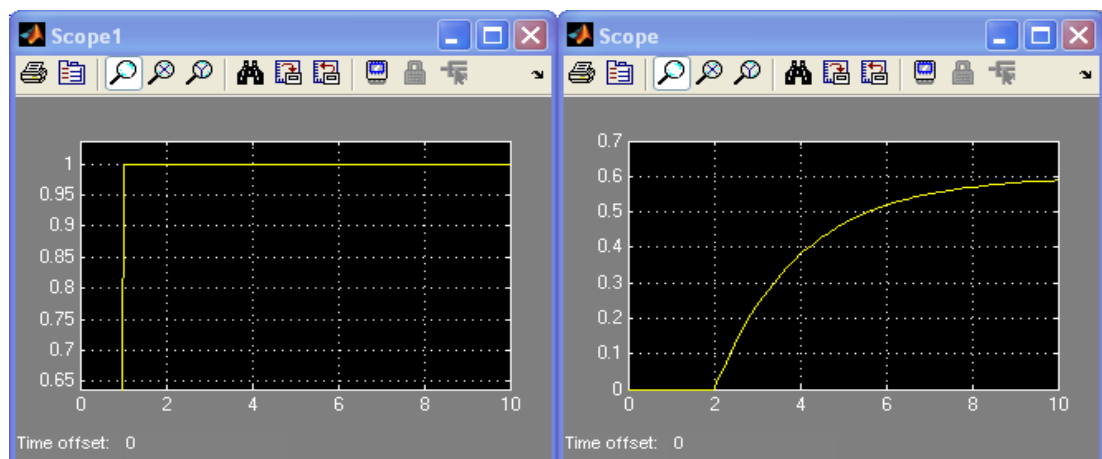


**Gambar B.10.** Jendela Parameter Blok fungsi aih

Hasil akhir yang nampak dari pengerjaan langkah-langkah diatas terlihat pada gambar B.11. Sedangkan gambar B.12 adalah salah satu hasil simulasi proses tersebut.



**Gambar B.11.** Blok Simulasi model proses persamaan B.1



**Gambar B.12.** Salah satu hasil simulasi model Simulink pada gambar B.11

ooOoo